THEORETICAL INVESTIGATION OF THE FLUTTER CHARACTERISTICS OF SUPERSONIC CASCADES WITH SUBSONIC LEADING-EDGE LOCUS

James Keith Bell

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

THEORETICAL INVESTIGATION OF THE FLUTTER CHARACTERISTICS OF SUPERSONIC CASCADES WITH SUBSONIC LEADING-EDGE LOCUS

by

James Keith Bell

June 1975

Thesis Advisor: M. F. Platzer

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*

THEORETICAL INVESTIGATION OF THE FLUTTER CHARACTERISTICS OF SUPERSONIC CASCADES WITH SUBSONIC LEADING-EDGE LOCUS

5. TYPE OF REPORT & PERIOD COVERED

Engineer's Thesis (June 1975)

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

James Keith Bell

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE

June 1975

13. NUMBER OF PAGES

193

14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*

Naval Postgraduate School
Monterey, California 93940

15. SECURITY CLASS. *(of this report)*

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

cascade; subsonic leading-edge locus; flutter

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Supersonic flow past a semi-infinite array of flat plate cascaded airfoils is analyzed using a linearized method of characteristics procedure. For a cascade having subsonic leading-edge locus, but otherwise arbitrary geometry, frequency of blade oscillation, and interblade phase angle, pressure distributions and lift forces and

Theoretical Investigation of the Flutter Characteristics of Supersonic Cascades with Subsonic Leading Edge Locus

by

James Keith Bell
Lieutenant, United States Navy
B.B.A., North Texas State University, 1966

Submitted in partial fulfillment of the

requirements for the degree of

AERONAUTICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
June 1975

moments are determined. These are used for flutter analysis and the determination of flutter boundaries. In addition, the linearized method of characteristics procedure is used to analyze supersonic flow through cylindrical shells. Pressure distributions and generalized forces are computed for arbitrary radius-to-length ratios, axial and circumferential mode numbers, and frequency of panel oscillation.

ABSTRACT

Supersonic flow past a semi-infinite array of flat plate cascaded airfoils is analyzed using a linearized method of characteristics procedure. For a cascade having subsonic leading edge locus, but otherwise arbitrary geometry, frequency of blade oscillation, and interblade phase angle, pressure distributions and lift forces and moments are determined. These are used for flutter analysis and the determination of flutter boundaries. In addition, the linearized method of characteristics procedure is used to analyze supersonic flow through cylindrical shells. Pressure distributions and generalized forces are computed for arbitrary radius-to-length ratios, axial and circumferential mode' numbers, and frequency of panel oscillation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

The following symbols are used in the CASCADE portion of the thesis.

$\bar{\alpha}$	Mach angle
α	sinusoidal pitch displacement
α_0	amplitude of blade pitch oscillation
β	stagger angle
β'	compatible stagger angle
γ	ratio of specific heats
δ_{crit}	the interblade phase angle at the minimum flutter speed
δ	interblade phase angle
ϵ	gridfineness ratio
ξ,η	characteristic directions
θ	temperature (Rankine)
μ	blade density parameter
π	3.141593
ρ_0,ρ	density (freestream, perturbation)
σ	cascade solidity (chord/blade spacing)
ω	circular frequency of oscillation
ω_α,ω_h	natural frequency of pitch,plunge (Eqs. IV-7)
	circular frequency at flutter
Ω_α	the quantity μr_α^2
Ω_h	the quantity $\mu(\frac{\omega_h}{\omega_\alpha})^2$
A_I	the quantity $\frac{1}{2}k\,\Delta x$
B_I	the quantity $\frac{1}{4}k\,(\frac{M^2}{M^2-1})\,\Delta x$

b	blade semi-chord
$\hat{c}$	blade chord
$\bar{c}, c_0, c$	sonic velocity (local, freestream, perturbation)
C, U, V	amplitude of the Teipel functions (sonic, axial, normal velocity)
d	distance between cascade blades
c_p, c_v	specific heats (constant pressure, volume)
g_α	structural damping coefficient
h	sinusoidal plunge displacement, positive down
h_0	amplitude of blade plunge oscillation
I	blade mass moment per unit span about the elastic axis
i	$\sqrt{-1}$
K_α, K_h	constants for the springs assumed to oppose the blade displacement in pitch and plunge
k_{crit}	the reduced frequency for which the minimum flutter speed occurs
k	reduced frequency
k_c	compressible reduced frequency, $k_c = k\frac{M}{M^2-1}$
L_h, L_α	complex lift coefficient for blade oscillation in plunge, pitch
L_1, L_2	components of lift coefficient due to plunge defined by Eqs. (IV-10a) and (IV-22a,b)
L_3, L_4	components of lift coefficient due to pitch defined by Eqs. (IV-10a) and (IV-22c,d)
M	Mach number

M_b	blade mass per unit span
M_h, M_α	complex moment coefficient for blade oscillation in plunge, pitch
M_1, M_2	components of moment coefficient resulting from plunge defined by Eqs. (IV-10b) and (IV-22e,f)
M_3, M_4	components of moment coefficient resulting from pitch defined by Eqs. (IV-10b) and (IV-22g,h)
n	one of a semi-infinite number of blades
N_{bld}	the n^{th} blade
$\bar{p}, p_o, p$	pressure (local, freestream, perturbation)
P_h, P_α	complex pressure distribution resulting from blade motion in plunge, pitch
R	gas constant for air
r_α	blade radius of gyration $(\frac{I_\alpha}{M_b b^2})$
s	entropy
S_b	blade static moment per unit span about the elastic axis
$\bar{u}, u_o, u$	axial velocity (local, freestream, perturbation)
u_{int}	internal energy of a gas
t	non-dimensional time
T	dimensional time $(T = t\frac{\hat{c}}{u_o})$

$\bar{V}$	velocity vector
v	normal velocity perturbation
x_o	elastic axis of the blade
x_α	location of the blade center of gravity measured from the elastic axis, positive aft $(\frac{S_b}{M_b b})$
x, y	non-dimensional coordinates
X, Y	dimensional coordinates ($X = \hat{c}x$, $Y = \hat{c}y$)
X	in the flutter section, the ratio of natural frequency in pitch to the circular frequency $(\frac{\omega_\alpha}{\omega})^2$
Y_u, Y_L	equation of the upper and lower surfaces of a cascade blade

The following symbols are used in the CYLINDRICAL SHELL portion of the thesis (Section VIII).

γ, δ	defined by Eqs. (VIII-31)
π	3.141593
Φ	unsteady velocity perturbation potential
ϕ	unsteady velocity perturbation potential amplitude
ϕ_1, ϕ_2	rate of change in ϕ along a left,right characteristic
ϕ_{12}, ϕ_{21}	cross derivatives of ϕ_1 and ϕ_2
ψ_m	panel deflection amplitude resulting from the m^{th} axial mode number
c	sonic velocity
$\bar{C}_p$	unsteady pressure coefficient

C_p	pressure coefficient amplitude
$\bar{h}$	time dependent panel surface displacement
i	$\sqrt{-1}$
k	the reduced frequency $(\frac{\omega L}{u_0})$
L	flexible panel length
M	Mach number
m	axial mode number
n	circumferential mode number
Q_{mr}	generalized force, explained in Section VIII-D
R_o	radius of the inner surface of the outer cylinder
R_i	radius of the outer surface of the inner cylinder
s, s_1, s_2	length along a characteristic, left, right
t	non-dimensional time
T	dimensional time, $T = t(\frac{L}{u_0})$
x, r, θ	non-dimensional cylindrical coordinates
X, R, θ	dimensional cylindrical coordinates ($X = Lx$, $R = Lr$, $\theta = \theta$)
z	axial deflection of panel surface
A,B,C	defined by Eqs. (VIII-24)
D,E,F	defined by Eqs. (VIII-25)
P,Q,S	grid point locations in the computational molecule

ACKNOWLEDGEMENT

This thesis was written with the close support and assistance of Associate Professor M. F. Platzer. Also, the author would like to thank Drs. J. Verdon (United Aircraft Research Laboratories), S. Fleeter and L. Snyder (Detroit Diesel Allison) for their constructive comments and suggestions, and Drs. J. Fagan (detroit Diesel Allison), R. Arnoldi (Pratt and Whitney Aircraft) and Mr. F. Carta (United Aircraft Research Laboratories) for their courtesy and consideration during his Navy funded experience tour at their respective facilities.

I. INTRODUCTION

The current design trend toward higher tip speed, reduced engine weight, and higher pressure ratios has made the modern high speed fan and compressor more susceptible to supersonic unstalled flutter. Unstalled flutter, as distinguished from stall flutter, occurs at small angle of attack and with attached flows. In addition, it occurs at or near the design condition. Flutter itself is a self-excited, unsteady phenomenon which can occur when the exciting forces just equal the elastic forces of the turbomachinery blades. For a review of the problem of fan and compressor flutter, refer to Ref. 6.

The original work in flutter analysis evolved from the flutter of aircraft wings, which was observed as early as 1914. Theodorsen [Ref. 22] developed the most basic analytic treatment of unsteady, subsonic aerodynamics for an airfoil oscillating in pitch and plunge. This analysis dealt with what has become known as classical flutter, and showed that flutter was only possible when the natural frequencies of flexure and torsion coalesced as the relative velocity over the wing was increased. Each mode of oscillation by itself (pitch or plunge) was stable.

Cascade flutter differs from classical flutter in that the inertia and stiffness properties of a turbomachine blade differ greatly from those of a wing. As a result, the critical flutter speed, the speed at which the onset of flutter occurs, is very much greater. Garrick and Rubinow [Ref. 9] extended the work of Theodorsen to include supersonic flow, and their analysis predicted that single degree of freedom torsional flutter was possible, although the bending mode was still stable. Lane [Ref. 14] showed

that cascade flutter could be analyzed by considering only a single oscillating blade.

Work on cascades with subsonic leading edge locus is fairly recent. The first attempt was made by Gorelov [Ref. 10]. Verdon [Ref. 23] obtained some of the first numerical results using a finite difference procedure. Brix and Platzer [Refs. 3,4] were also able to obtain results using a method of characteristics approach. Another solution was presented by Nagashima and Whitehead [Ref. 15] using dipole distributions. All three solutions were in close agreement. Sisto and Ni [Ref. 20] developed a 'time marching" technique applicable to an infinite cascade. Their results appear to be in good agreement with the finite cascade results.

An analytic development, valid for low frequency blade motion, has been developed by Kurosaka [Ref. 11] and is currently being extended to the higher frequency range [Ref. 12]. These results are applied to an infinite cascade. Very recently, another analytic treatment has been presented by Verdon [Ref. 25] (see also Verdon and McCune [Ref. 24]) in which an infinite series representation is used to express the influence of neighboring blades and wakes on the reference passage. However, series convergence was not always obtained, and some doubt exists as to its uniform validity [Ref. 13]. A new analysis currently being developed by Verdon, applicable to the infinite cascade, does appear to yield uniformly good results [Ref. 26]. Actual wind tunnel tests of cascade flutter have been made by Snyder and Commerford [Ref. 19] and by Fleeter [Ref. 7].

In the current study, a linearized method of characteristics procedure, based on a developement by Teipel [Ref. 21] for unsteady aerodynamics of a flat plate airfoil, is used to examine the pressure distributions and

flutter boundaries for cascade blades having subsonic leading edge locus. It is an extension of previous work done by Chalkley [Ref. 5] and Brix [Ref. 2].

In addition, the method of characteristics was applied to the problem of supersonic flow through oscillating cylindrical shells to determine the pressure distribution and associated generalized forces acting on the cylinder walls.

Although it is unreasonable to suppose that the two-dimensional cascade investigation can accurately model the complex mixed transonic-supersonic flow of an actual present day turbomachine, it is hoped that this analysis will provide a better understanding of the problems involved, and serve as a starting point for the flutter analysis during the design stage of modern high speed fans and compressors.

II. THEORY

A. PROBLEM FORMULATION

Consider a cascade composed of a semi-infinite array of thin, two-dimensional flat plate airfoils immersed in the supersonic flow of an inviscid, non-conducting, ideal gas having constant specific heats. Additionally, the flow field is assumed to be irrotational and isentropic.

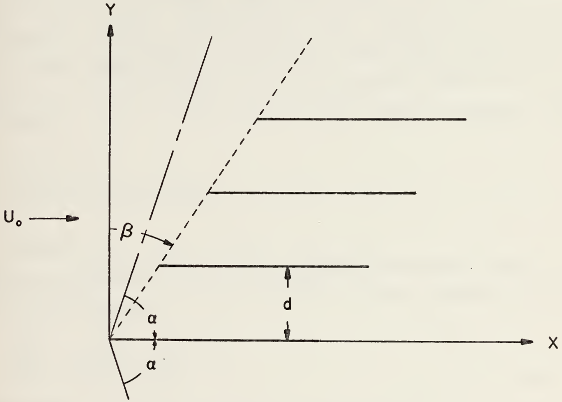


Figure 1. Cascade geometry

The only restriction on the cascade is that the locus of blade leading edges must be subsonic; i.e.,

$$\beta > \frac{\pi}{2} - \bar{\alpha} \qquad \text{(II- 1)}$$

where β is the stagger angle and $\bar{\alpha}$ is the Mach angle ($\sin^{-1}\frac{1}{M}$). As shown in figure 1, the cascade is aligned in the X-Y coordinate system so that the leading-edge of the first blade is at the origin and the blade itself lies along the X-axis.

Perturbations in the flow field are caused by blade motion in pitch and plunge, assumed to be of small amplitude with simple harmonic motion. For a cascade with subsonic leading-edge locus, the blades are swept aft of the initial Mach wave. Thus, disturbances are allowed to propagate throughout the domain of influence (the region downstream of the Mach cone emanating from the leading edge of the first blade), and are not restricted to the region between blade passages.

B. GOVERNING EQUATIONS

Subject to the initial assumptions stated above, the governing equations for this boundary value problem are the continuity equation

$$\frac{D\rho}{DT} + \rho\left(\frac{\partial u}{\partial X} + \frac{\partial v}{\partial Y}\right) = 0 \qquad \text{(II- 2)}$$

the equation of motion (Euler equations)

$$\frac{Du}{DT} + \frac{1}{\rho}\frac{\partial p}{\partial X} = 0 \qquad \text{(II-3a)}$$

$$\frac{Dv}{DT} + \frac{1}{\rho}\frac{\partial p}{\partial Y} = 0 \qquad \text{(II-3b)}$$

the condition of irrotationality

$$\frac{\partial u}{\partial Y} - \frac{\partial v}{\partial X} = 0 \qquad \text{(II- 4)}$$

and the energy equation

$$\frac{Ds}{DT} = 0 \qquad \text{(II- 5)}$$

$\frac{D}{DT}$ denotes the material, or substantive, derivative of a fluid particle which is a function of position and time. It is composed of the local derivative (rate of change with respect to time) and the convective derivative (rate of change with respect to position) and is written as

$$\frac{D(\ )}{DT} = \frac{\partial(\ )}{\partial T} + (\bar{V}\cdot\quad)(\ ) \qquad \text{(II-5a)}$$

where $\bar{V}$ is the velocity vector. The local sonic velocity is given by,

$$\bar{c}^2 = \left(\frac{\partial p}{\partial \rho}\right)_s = \frac{dp}{d\rho} \qquad \text{(II- 6)}$$

and

$$\frac{p}{\rho^{\gamma}} = \text{constant} \qquad \text{(II- 7)}$$

Taking the total differential of Eq. (7)

$$\left(\frac{1}{\rho}\right)^{\gamma} dp - \gamma p \left(\frac{1}{\rho}\right)^{(\gamma+1)} d\rho = 0 \qquad \text{(II- 8)}$$

Rearranging

$$\frac{dp}{d\rho} = \gamma \frac{p}{\rho} \qquad \text{(II- 9)}$$

or

$$\bar{c}^2 = \gamma \frac{p}{\rho} \qquad \text{(II-10)}$$

C. BOUNDARY CONDITIONS

The flow boundary condition (flow tangency) for a blade having an equation of the form

$$F(X,Y,T) = 0 \qquad \text{(II-11)}$$

is developed as follows. At each point of a solid-fluid surface, at every instant, the normal component of the relative velocity between the solid and the fluid must vanish. In other words, the total time rate of change in F(X,Y,T) following a surface particle of the solid is zero. Mathematically, this statement is expressed as

$$\frac{DF}{DT} = 0 \qquad \text{(II-12)}$$

For the cascade, the equations for the upper and lower surfaces of a blade are written

$$F_u(X,Y,T) = Y - Y_u(X,T) = 0 \qquad \text{(II-13)}$$

and

$$F_L(X,Y,T) = Y - Y_L(X,T) = 0 \qquad \text{(II-14)}$$

Applying Eq. (12) to Eqs. (13) and (14)

$$\frac{DF_u}{DT} = \frac{\partial F_u}{\partial T} + \bar{u}\frac{\partial F_u}{\partial X} + v\frac{\partial F_u}{\partial Y} = 0 \qquad \text{(II-15)}$$

or

$$v - \frac{\partial Y_u}{\partial T} - \bar{u}\frac{\partial Y_u}{\partial X} = 0 \qquad \text{(II-16)}$$

since $\frac{\partial F_u}{\partial Y} = 1$. Likewise,

$$\frac{DF_L}{DT} = v - \frac{\partial Y_L}{\partial T} - \bar{u}\frac{\partial Y_L}{\partial X} = 0 \qquad \text{(II-17)}$$

The other boundary condition that must be enforced is the Sommerfeld radiation condition, i.e., disturbances must propagate away from their sources.

D. LINEARIZATION

The normal velocity perturbation is written

$$v(X,Y_u(X,T)) = \frac{\partial Y_u}{\partial T} + \bar{u}\frac{\partial Y_u}{\partial X} \qquad \text{(II-18)}$$

$$v(X,Y_L(X,T)) = \frac{\partial Y_L}{\partial T} + \bar{u}\frac{\partial Y_L}{\partial X} \qquad \text{(II-19)}$$

From the assumption of thin blades and small amplitude oscillations, all flow quantities may be considered to be small quantities which are linearly superimposed onto the freestream quantities. Thus

$$\bar{u} = u + u_0 \qquad \text{(II-20)}$$

$$\bar{c} = c + c_0 \qquad \text{(II-21)}$$

$$v = v \qquad \text{(II-22)}$$

while the density and pressure perturbations are

$$\Delta p = p - p_0 \qquad \text{(II-23)}$$

$$\Delta\rho = \rho - \rho_o \tag{II-24}$$

Also, because of the assumption of thin blades, Y_u and Y_L are small quantities with respect to axial distances along the blade. Thus, substituting Eqs. (20) and (22) into Eqs. (18) and (19) and neglecting the effect of higher order terms, the normal velocity perturbation equation becomes

$$v(X,Y_u(X,T)) = \frac{\partial Y_u}{\partial T} + u_o\frac{\partial Y_u}{\partial X} \tag{II-25}$$

$$v(X,Y_L(X,T)) = \frac{\partial Y_L}{\partial T} + u_o\frac{\partial Y_L}{\partial X} \tag{II-26}$$

To transfer the boundary condition to the axis, Eqs. (25) and (26) are expanded in a Taylor series about $Y = 0$.

$$v(X,Y_u(X,T)) = v(X,Y=0^+,T) + Y_u\frac{\partial v(X,Y=0^+,T)}{\partial Y} + \frac{1}{2}Y_u^2\frac{\partial^2 v(X,Y=0^+,T)}{(\partial Y)^2} + \cdots \tag{II-27}$$

$$v(X,Y_L(X,T)) = v(X,Y=0^-,T) + Y_L\frac{\partial v(X,Y=0^-,T)}{\partial Y} + \frac{1}{2}Y_L^2\frac{\partial^2 v(X,Y=0^-,T)}{(\partial Y)^2} + \cdots \tag{II-28}$$

Neglecting higher order terms,

$$v(X,Y_u(X,T)) = v(X,Y=0^+,T) \tag{II-29}$$

$$v(X,Y_L(X,T)) = v(X,Y=0^-,T) \tag{II-30}$$

and thus

$$v(X,0^+,T) = \frac{\partial Y_u}{\partial T} + u_o \frac{\partial Y_u}{\partial X} \qquad \text{(II-31)}$$

$$v(X,0^-,T) = \frac{\partial Y_L}{\partial T} + u_o \frac{\partial Y_L}{\partial X} \qquad \text{(II-32)}$$

For cascaded blades, it is assumed that the n^{th} blade leads the motion of the $(n-1)^{st}$ blade by an amount δ, the interblade phase angle. Expressing this and the assumption of sinusoidal motion, the equation for the upper and lower blade surfaces becomes

$$Y_u(X,T) = Y_u(X)\, e^{i[\omega T + (n-1)\delta]} \qquad \text{(II-33)}$$

$$Y_L(X,T) = Y_L(X)\, e^{i[\omega T + (n-1)\delta]} \qquad \text{(II-34)}$$

which reduces the flow tangency condition, Eqs. (31) and (32), to

$$v(X,0^+,T) = [i\omega Y_u(X) + u_o \frac{\partial Y}{\partial X}] e^{i\{(n-1)\delta\}} e^{i\omega T} \qquad \text{(II-35)}$$

$$v(X,0^-,T) = [i\omega Y_L(X) + u_o \frac{\partial Y}{\partial X}] e^{i\{(n-1)\delta\}} e^{i\omega T} \qquad \text{(II-36)}$$

Now let $Y_u(X,T) = Y_L(X,T) = \bar{Y}(X) e^{i\omega T}$. Then the flow tangency condition for pitch is,

$$\bar{Y}(X)_{pitch} = -\alpha_o (X - x_o \hat{c}) \qquad \text{(II-37)}$$

where,

$$\alpha = \alpha_o e^{i\omega T} \tag{II-38}$$

while for the plunge,

$$\bar{Y}(X)_{plunge} = - h_o \hat{c} \tag{II-39}$$

where,

$$h = h_o e^{i\omega T} \tag{II-40}$$

Deflections are defined as positive downward, and angle of twist positive leading edge up. h_o and α_o are dimensionless complex amplitudes, with x_o the twist axis location (elastic axis) in percent chord $\hat{c}$. Figure 2 shows the positive direction of all quantities used in defining the flow tangency condition.

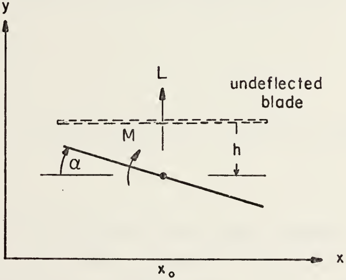


Figure 2. Positive directions of Lift and Moment resulting from deflections in pitch (α) and/or plunge (h).

To non-dimensionalize the flow tangency equations, distances are scaled to the blade chord and time to blade

chord divided by freestream velocity u_0. This leads to the introduction of the reduced frequency parameter

$$k = \frac{\omega c}{u_o} \qquad \text{(II-41)}$$

which is twice the reduced frequency as defined by Garrick and Rubinow [Ref. 9] who scaled all quantities to blade semi-chord.

Non-dimensionalizing Eqs. (35) and (36) and using the definition of reduced frequency above, the flow tangency conditions become

pitch:

$$\frac{v(x,0\pm,t)}{u_o} = -\alpha_0\{\cos[(n-1)\delta] - k(x-x_0)\sin[(n-1)\delta]\}e^{ikt}$$
$$- i\alpha_0\{k\sin[(n-1)\delta] + k(x-x_0)\cos[(n-1)\delta]\}e^{ikt} \qquad \text{(II-42)}$$

plunge:

$$\frac{v(x,0\pm,t)}{u_o} = h_0\{k\sin[(n-1)\delta] - ik\cos[(n-1)\delta]\}e^{ikt} \qquad \text{(II-43)}$$

To linearize the expression for the pressure, Eqs. (10), (20) and (24) are used. Expand Eq. (10) in terms of perturbation quantities. Through Eq. (48), second order terms are neglected whenever they occur.

$$c_0^2 + 2cc_0 = \gamma\frac{(p_o + \Delta p)}{(1 + \frac{\Delta\rho}{\rho_o})} \qquad \text{(II-44)}$$

As $\frac{\Delta\rho}{\rho_o} < 1$, the denominator can be expanded in a geometric

series of the form

$$\frac{1}{1+\epsilon} = 1 - \epsilon + \epsilon^2 - \epsilon^3 + \cdots \qquad \text{(II-45)}$$

to yield,

$$c_0^2 + 2cc_0 = \gamma(p_0 + \Delta p)\,(1 - \frac{\Delta\rho}{\rho_0}) \qquad \text{(II-46)}$$

Expanding,

$$c_0^2 + 2cc_0 = (1 - \frac{c_0^2}{\gamma}\frac{\Delta\rho}{\Delta p})\,\Delta p \qquad \text{(II-47)}$$

However, as Δp and $\Delta\rho$ are both small quantities

$$\frac{\Delta\rho}{\Delta p} = (\frac{dp}{d\rho})^{-1} = \frac{1}{\bar{c}^2} \qquad \text{(II-48)}$$

Expanding Eq. (47), substituting into Eq. (46) and rearranging terms,

$$\frac{\Delta p}{\rho_0} = \frac{2}{\gamma-1}cc_0\,[1 + \frac{2\gamma c}{(\gamma-1)\,c_0}]^{-1} \qquad \text{(II-49)}$$

But $\frac{2\gamma c}{(\gamma-1)\,c_0} < 1$. Using Eq. (43) to expand the term in brackets yields the linearized form of the pressure difference

$$p - p_0 = \frac{2}{\gamma-1}\rho_0 c_0(\bar{c} - c_0) \qquad \text{(II-50)}$$

III. METHOD OF CHARACTERISTICS

A. CHARACTERISTIC DIRECTIONS

To introduce the method of characteristics, it is important to recall that for supersonic flow there are distinct regions of influence and dependence. Hence, curves must exist along which the values of the flow variables and their derivatives are known, yet the flow field in the immediate neighborhood of these curves is indeterminate. Therefore, to obtain the directions in the x-y plane of these curves, which are called characteristic directions, or more simply, characteristics, one must prescribe that the derivatives across these curves are indeterminate.

Mathematically, this means that the dependent variables must be continuous, but their first, or higher order, derivatives may be discontinuous. Physically, the characteristics act as information carriers, in that all disturbances propagate along them. In particular, they represent the paths of Mach waves, and are therefore referred to as Mach lines.

To facilitate the analysis, it is desirable to introduce new coordinates ξ and η. The governing equations are written in terms of the two arbitrary, but intersecting curves

$$\xi = \xi(x,y) = \text{constant} \tag{III-1a}$$

and

$$\eta = \eta(x,y) = \text{constant} \tag{III-1b}$$

However, the equations are more suitably transformed to the arbitrary coordinates if the dependent variables are $\bar{u}$, v

and $\bar{c}$. Here, air is assumed to be a perfect gas with constant specific heats c_p (constant pressure) and c_v (constant volume). Rewriting Eq. (II-10) as

$$\bar{c}^2 = \gamma R\theta \qquad \text{(III- 2)}$$

$$R = c_p - c_v \qquad \text{(III- 3)}$$

where R is the gas constant for air, and θ is temperature in degrees Rankine. Taking the total differential of Eq. (2) and dividing by $\bar{c}^2$,

$$2\frac{d\bar{c}}{\bar{c}} = \frac{d\theta}{\theta} \qquad \text{(III- 4)}$$

Taking the total differential of Eq. (II-10),

$$2\bar{c}d\bar{c} = \gamma\frac{1}{\rho}dp - p\frac{d\rho}{\rho^2} \qquad \text{(III- 5)}$$

or, dividing by $\bar{c}^2$,

$$2\frac{d\bar{c}}{\bar{c}} = \frac{dp}{p} - \frac{d\rho}{\rho} \qquad \text{(III- 6)}$$

Now from the First Law of Thermodynamics and the fact that the flow is homentropic,

$$\theta ds = du_{int} - \frac{p}{\rho^2}d\rho \qquad \text{(III- 7)}$$

For an ideal gas, $du_{int} = c_v d\theta$ and Eq. (7) may be written as

$$\theta ds = c_v d\theta - R\theta\frac{d\rho}{\rho} \qquad \text{(III- 8)}$$

Dividing by $c_v\theta$

$$\frac{1}{c_v}ds = \frac{d\theta}{\theta} - (\gamma-1)\frac{d\rho}{\rho} \qquad \text{(III- 9)}$$

and using Eqs. (4) and (6) yields

$$\frac{1}{c_v}ds = 2\frac{d\bar{c}}{\bar{c}} - (\gamma-1)\frac{1}{p}dp \qquad \text{(III-10)}$$

But because the thermodynamic properties of a continuous flow field are to be considered as continuous, single-valued scalar point functions, Eq. (10) must hold for any change of the fluid properties s, $\bar{c}$ and p. Hence, it may be written in terms of material derivatives. Then, subject to Eq. (II-5), Eq. (10) becomes

$$\frac{1}{c_v}\frac{Ds}{DT} = 2\frac{1}{\bar{c}}\frac{D\bar{c}}{DT} - (\gamma-1)\frac{1Dp}{pDT} = 0 \qquad \text{(III-11)}$$

Then, using Eq. (II-10), Eq. (11) reduces to

$$\frac{Dp}{DT} = \frac{2}{\gamma-1}\bar{c}\frac{D\bar{c}}{DT} \qquad \text{(III-12)}$$

Now the perturbation equations are introduced. Writing Eq. (II-9) as

$$\frac{Dp}{DT} = \bar{c}^2\frac{D\rho}{DT} \qquad \text{(III-13)}$$

and substituting into Eq. (12) allows Eq. (II-2) to be expressed as

$$\frac{2}{\gamma-1}\frac{\partial\bar{c}}{\partial T} + \frac{2}{\gamma-1}u\frac{\partial\bar{c}}{\partial X} + c_o\frac{\partial u}{\partial X} + c_o\frac{\partial v}{\partial Y} = 0 \qquad \text{(III-14)}$$

neglecting the effect of higher order terms. This is the form of the continuity equation which is to be used in the transformation.

To render the first Euler equation, Eq. (II-3), in a form suitable for transformation, use Eq. (11) in differential form

$$2\frac{\gamma}{c}d\bar{c} = (\gamma-1)\frac{1}{p}dp \qquad \text{(III-15)}$$

With Eq. (II-9) this yields

$$\frac{1}{\rho}dp = \frac{2}{\gamma-1}\bar{c}d\bar{c} \qquad \text{(III-16)}$$

Expanding the total derivatives and equating the partial derivatives, one obtains

$$\frac{1}{\rho}\frac{\partial p}{\partial X} = \frac{2}{\gamma-1}\bar{c}\frac{\partial \bar{c}}{\partial X} \qquad \text{(III-17)}$$

Substituting into the first Euler equation

$$\frac{D\bar{u}}{DT} = \frac{\partial \bar{u}}{\partial T} + \bar{u}\frac{\partial \bar{u}}{\partial X} + v\frac{\partial \bar{u}}{\partial Y} = -\frac{2}{\gamma-1}\bar{c}\frac{\partial \bar{c}}{\partial X} \qquad \text{(III-18)}$$

Introducing the small perturbation quantities and neglecting second order quantities yields the desired form of the first Euler equation,

$$\frac{\partial \bar{u}}{\partial T} + u_o\frac{\partial \bar{u}}{\partial X} + \frac{2}{\gamma-1}c_o\frac{\partial \bar{c}}{\partial X} = 0 \qquad \text{(III-19)}$$

The condition of irrotationality

$$\frac{\partial v}{\partial Y} - \frac{\partial \bar{u}}{\partial X} = 0 \qquad \text{(III-20)}$$

is used in place of the second Euler equation as one of the governing equations to be transformed. Then the continuity equation, the first Euler equation and the condition of irrotationality form a system of three equations in the three unknowns $\bar{u}$, v and $\bar{c}$. For the assumption of simple

harmonic motion, the Teipel amplitude functions [Ref. 21] are introduced,

$$U(X,Y)e^{i\omega T} = \frac{\bar{u} - u_o}{u_o} \qquad \text{(III-21a)}$$

$$V(X,Y)e^{i\omega T} = \frac{1}{\sqrt{M^2-1}}\frac{v}{u_o} \qquad \text{(III-21b)}$$

$$C(X,Y)e^{i\omega T} = \frac{2}{\gamma-1}(\frac{1}{M^2})\frac{\bar{c} - c_o}{c_o} \qquad \text{(III-21c)}$$

where U, V and C are complex non-dimensional amplitudes. U and V are the perturbation velocities and C is the perturbation in sonic velocity which will be related to the pressure coefficient through Eq. (II-49).

Substituting Eqs. (21) into Eqs. (14), (19) and (20), and then non-dimensionalizing as before yields,

$$\frac{\partial U}{\partial x} + \sqrt{M^2-1}\,\frac{\partial V}{\partial y} + M^2\frac{\partial C}{\partial x} + ikM^2C = 0 \qquad \text{(III-22a)}$$

$$\frac{\partial U}{\partial x} + \frac{\partial C}{\partial x} + ikU = 0 \qquad \text{(III-22b)}$$

$$\frac{\partial U}{\partial y} - \sqrt{M^2-1}\,\frac{\partial V}{\partial x} = 0 \qquad \text{(III-22c)}$$

For ξ = constant or η = constant, both $\xi(x,y(x))$ and $\eta(x,y(x))$ are implicit functions of x. Hence,

$$d\xi = \xi_x dx + \xi_y dy = 0 \qquad \text{(III-23a)}$$

and

$$d\eta = \eta_x dx + \eta_y dy = 0 \qquad \text{(III-23b)}$$

Therefore,

$$\frac{\xi_x}{\xi_y} = \frac{\eta_x}{\eta_y} = - \frac{dy}{dx} \qquad \text{(III- 24)}$$

and the labeling of the characteristic directions is arbitrary. By convention the left-running Mach line is normally denoted ξ and the right-running Mach line, η.

To solve for the characteristic directions in the x-y plane, Eqs. (22) are rewritten along lines of constant ξ and η.

$$\xi_x U_\xi + \sqrt{M^2-1}\, \xi_y V_\xi + M^2 \xi_x C_\xi = - \eta_x U_\eta - M^2 \eta_x C_\eta - \sqrt{M^2-1}\, \eta_y V_\eta - ikM^2 C \qquad \text{(III-25a)}$$

$$\xi_x U_\xi + \xi_x C_\xi = - \eta_x U - \eta_x C_\eta - ikU \qquad \text{(III-25b)}$$

$$\xi_y U - \sqrt{M^2-1}\, \xi_x V = - \eta_y U_\eta + \sqrt{M^2-1}\, \eta_x V_\eta \qquad \text{(III-25c)}$$

Now Cramer's rule is applied to solve for the derivatives of the dependent variables with respect to ξ along η = constant. To satisfy the condition that these derivatives be finite requires that each be of the form $\frac{0}{0}$, i.e., indeterminate. Equating the determinant of the coefficient matrix of $\{ U , C , V \}^t$ to zero yields,

$$\xi_x [\xi_x^2 (M^2-1) - \xi_y^2] = 0 \qquad \text{(III- 26)}$$

to which there are three real solutions, each of which defines a characteristic direction in the x-y plane. These are,

$$\frac{dy}{dx} = -\xi_x = 0 \tag{III-27a}$$

$$\frac{dy}{dx} = -\frac{\xi_x}{\xi_y} = \pm\frac{1}{\sqrt{M^2-1}} \tag{III-27b}$$

The same results are obtained when one solves for the first derivatives with respect to η along lines of ξ = constant.

Eq. (27a) gives the streamline slope in the flow field, while Eq. (27b) gives the slope of the two curves ξ = constant and η = constant. It should be noted that these are the same characteristic directions as obtained for steady, two-dimensional flow. By convention,

$$\left(\frac{dy}{dx}\right)_{\xi=\text{const}} = +\frac{1}{\sqrt{M^2-1}} \tag{III-28a}$$

$$\left(\frac{dy}{dx}\right)_{\eta=\text{const}} = -\frac{1}{\sqrt{M^2-1}} \tag{III-28b}$$

$$\left(\frac{dy}{dx}\right)_{\text{stream}} = 0 \tag{III-28c}$$

For an arbitrary function f = f(x,y), the total derivative is

$$df = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy \tag{III- 29}$$

For f(x,y) taken along a curve as defined by Eqs. (1), however, y is an implicit function of x and Eq. (29) is rewritten as

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{dy}{dx} \qquad \text{(III- 30)}$$

Therefore, along a streamline,

$$\left(\frac{df}{dx}\right)_{\text{stream}} = \frac{\partial f}{\partial x} \qquad \text{(III- 31)}$$

while along the Mach lines,

$$\left(\frac{df}{dx}\right)_{\xi=\text{const}} = \frac{\partial f}{\partial x} + \frac{1}{\sqrt{M^2-1}}\frac{\partial f}{\partial y} \qquad \text{(III-32a)}$$

$$\left(\frac{df}{dx}\right)_{\eta=\text{const}} = \frac{\partial f}{\partial x} - \frac{1}{\sqrt{M^2-1}}\frac{\partial f}{\partial y} \qquad \text{(III-32b)}$$

Then, solving for the partial derivatives in terms of the ordinary derivatives,

$$\frac{\partial f}{\partial x} = \left(\frac{df}{dx}\right)_{\text{str}} \qquad \text{(III-33a)}$$

$$\frac{\partial f}{\partial x} = \frac{1}{2}\left[\left(\frac{df}{dx}\right)_{\xi=\text{const}} + \left(\frac{df}{dx}\right)_{\eta=\text{const}}\right] \qquad \text{(III-33b)}$$

$$\frac{\partial f}{\partial y} = \frac{1}{2}\sqrt{M^2-1}\left[\left(\frac{df}{dx}\right)_{\xi=\text{const}} - \left(\frac{df}{dx}\right)_{\eta=\text{const}}\right] \qquad \text{(III-33c)}$$

The compatibility relations are obtained by writing Eqs. (22) using Eqs. (33). Thus,

$$\frac{1}{2}\left[\left(\frac{dU}{dx}\right)_{\xi} + \left(\frac{dU}{dx}\right)_{\eta}\right] + \frac{1}{2}(M^2-1)\left[\left(\frac{dV}{dx}\right)_{\xi} - \left(\frac{dV}{dx}\right)_{\eta}\right]$$

$$+ \frac{M^2}{2}\left[\left(\frac{dC}{dx}\right)_{\xi} + \left(\frac{dC}{dx}\right)_{\eta}\right] + ikM^2C = 0 \qquad \text{(III-34a)}$$

$$\frac{1}{2}\left[\left(\frac{dU}{dx}\right)_{\xi} + \left(\frac{dU}{dx}\right)_{\eta}\right] + \frac{1}{2}\left[\left(\frac{dC}{dx}\right)_{\xi} + \left(\frac{dC}{dx}\right)_{\eta}\right] + ikU = 0 \tag{III-34b}$$

$$\left[\left(\frac{dU}{dx}\right)_{\xi} - \left(\frac{dU}{dx}\right)_{\eta}\right] - \left[\left(\frac{dV}{dx}\right)_{\xi} + \left(\frac{dV}{dx}\right)_{\eta}\right] = 0 \tag{III-34c}$$

Eq. (34b) is multiplied by M^2 and then subtracted from Eq. (34a). The resulting equation is first added to, and then subtracted from Eq. (34c) to give two of the compatibility equations

$$\left(\frac{dU}{dx}\right)_{\xi} - \left(\frac{dV}{dx}\right)_{\xi} + ik\frac{M^2}{M^2-1}(U - C) = 0 \tag{III-35a}$$

$$\left(\frac{dU}{dx}\right)_{\eta} + \left(\frac{dV}{dx}\right)_{\eta} + ik\frac{M^2}{M^2-1}(U - C) = 0 \tag{III-35b}$$

The third compatibility relation is obtained from writing Eq. (34b) along a streamline.

$$\left(\frac{dU}{dx}\right)_{\text{stream}} + \left(\frac{dC}{dx}\right)_{\text{stream}} + ikU = 0 \tag{III-35c}$$

The governing equations have now been reduced to a system of three ordinary differential equations which are implicitly a function of x, with U , V and C complex amplitudes. Thus, Eqs. (35) are really six equations, with three real and three imaginary components.

To write the boundary conditions in terms of the Teipel amplitude functions, note that

$$v(x,0,t) = u_0\sqrt{M^2-1}\,V(x,y)e^{ikt} \tag{III- 36}$$

from the definition of the amplitude function, and therefore the flow tangency condition is,

$$V = V(x,y) = \frac{1}{\sqrt{M^2-1}} \frac{v(x,0,t)}{u_o} \qquad \text{(III- 37)}$$

where $\frac{v(x,0,t)}{u_o}$ is given by (II-40) for pitch and (II-41) for plunge.

B. FINITE DIFFERENCE EQUATIONS

To write Eqs. (III-34) in finite difference form, the computational molecule shown in figure 3 is used.

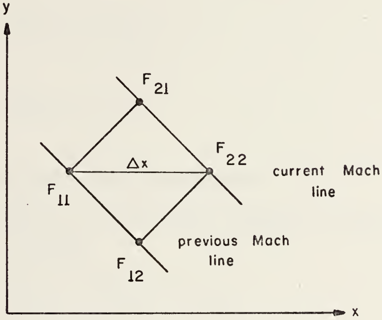


Figure 3. Computational molecule for a general flow field point.

If F is an arbitrary complex flow quantity, derivatives with respect to x along the characteristics, in finite difference form, are written,

$$\left(\frac{dF}{dx}\right)_{str} = \frac{F_{22} - F_{11}}{\Delta x} \qquad \text{(III-38a)}$$

$$\left(\frac{dF}{dx}\right)_{\xi} = \frac{F_{22} - F_{12}}{\frac{1}{2}\Delta x} \qquad \text{(III-38b)}$$

$$\left(\frac{dF}{dx}\right)_{\eta} = \frac{F_{22} - F_{21}}{\frac{1}{2}\Delta x} \qquad \text{(III-38c)}$$

The values of the perturbation quantities in Eqs. (38) are averages, so that

$$(F)_{str} = \frac{1}{2}(F_{11} + F_{22}) \qquad \text{(III-39a)}$$

$$(F)_{\xi} = \frac{1}{2}(F_{12} + F_{22}) \qquad \text{(III-39b)}$$

$$(F)_{\eta} = \frac{1}{2}(F_{22} + F_{21}) \qquad \text{(III-39c)}$$

1. General Flow Field Equations

With these relations substituted into Eqs. (35), the system of equations becomes,

$$U_{22R} + C_{22R} - A_I U_{22I} = K_{12R} \qquad \text{(III-40a)}$$

$$U_{22I} + C_{22I} + A_I U_{22R} = K_{12I} \qquad \text{(III-40b)}$$

$$U_{22R} - V_{22R} - B_I(U_{22I} - C_{22I}) = K_{34R} \qquad \text{(III-40c)}$$

$$U_{22I} - V_{22I} + B_I(U_{22R} - C_{22R}) = K_{34I} \qquad \text{(III-40d)}$$

$$U_{22R} + V_{22R} - B_I(U_{22I} - C_{22I}) = K_{56R} \qquad \text{(III-40e)}$$

$$U_{22I} + V_{22I} + B_I (U_{22R} - C_{22R}) = K_{56I} \tag{III-40f}$$

where

$$K_{12R} = U_{11R} + C_{11R} + A_I U_{11I} \tag{III-40g}$$

$$K_{12I} = U_{11I} + C_{11I} - A_I U_{11R} \tag{III-40h}$$

$$K_{34R} = U_{12R} - V_{12R} + B_I (U_{12I} - C_{12I}) \tag{III-40i}$$

$$K_{34I} = U_{12I} - V_{12I} - B_I (U_{12R} - C_{12R}) \tag{III-40j}$$

$$K_{56R} = U_{21R} + V_{21R} + B_I (U_{21I} - C_{21I}) \tag{III-40k}$$

$$K_{56I} = U_{21I} + V_{21I} - B_I (U_{21R} - C_{21R}) \tag{III-40l}$$

and

$$A_I = \frac{1}{2} k \Delta x \tag{III-41a}$$

$$B_I = \frac{1}{4} k \left(\frac{M^2}{M^2-1}\right) \Delta x \tag{III-41b}$$

As shown by Teipel [Ref. 21], solving these equations gives

$$U_{22R} = \frac{(1-A_I B_I)\left[\frac{1}{2}(K_{34R}+K_{56R}) - B_I K_{12I}\right] + 2B_I\left[\frac{1}{2}(K_{34I}+K_{56I}) + B_I K_{12R}\right]}{(1-A_I B_I)^2 + (2B_I)^2} \tag{III-42a}$$

$$U_{22I} = \frac{(1-A_I B_I)\left[\frac{1}{2}(K_{34I}+K_{56I}) - B_I K_{12R}\right] - 2B_I\left[\frac{1}{2}(K_{34R}+K_{56R}) - B_I K_{12I}\right]}{(1-A_I B_I)^2 + (2B_I)^2} \tag{III-42b}$$

$$V_{22R} = \frac{1}{2}(K_{56R} - K_{34R}) \tag{III-42c}$$

$$V_{22I} = \frac{1}{2}(K_{56I} - K_{34I}) \qquad \text{(III-42d)}$$

$$C_{22R} = K_{12R} - U_{22R} + A_I U_{22I} \qquad \text{(III-42e)}$$

$$C_{22I} = K_{12I} - U_{22I} - A_I U_{22R} \qquad \text{(III-42f)}$$

These are the finite difference equations for a grid point located in the general flow field.

2. Upper Surface of a Blade

For a grid point on top of a blade, the computational molecule of figure 3 is modified as shown in figure 4.

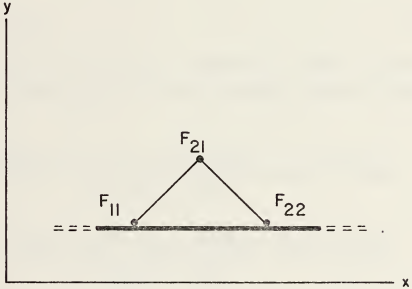


Figure 4. Computational molecule for a grid point on the upper surface of a blade.

Perturbation quantities at grid point F_{12} are considered zero and the normal velocity V_{22} (pitch or plunge at grid point F_{22}) is prescribed by the blade movement as given by

Eq. (II-37). Here the applicable equations are,

$$U_{22R} + C_{22R} - A_I U_{22I} = K_{12R} \tag{III-43a}$$

$$U_{22I} + C_{22I} + A_I U_{22R} = K_{12I} \tag{III-43b}$$

$$U_{22R} - B_I U_{22I} + B_I C_{22I} = K_{56R} - K_{34R} \tag{III-43c}$$

$$U_{22I} + B_I U_{22R} - B_I C_{22R} = K_{56I} - K_{34I} \tag{III-43d}$$

where K_{12} and K_{56} are as before, but

$$K_{34R} = V_{22R} \tag{III-43e}$$

$$K_{34I} = V_{22I} \tag{III-43f}$$

as given by the flow tangency condition. Blade position is always given relative to the leading edge of each blade. Solving,

$$U_{22R} = \frac{(1-A_I B_I)\left[K_{56R} - K_{34R} - B_I K_{12I}\right] + 2B_I\left[K_{56I} - K_{34I} + B_I K_{12R}\right]}{(1-A_I B_I)^2 + (2B_I)^2} \tag{III-44a}$$

$$U_{22I} = \frac{(1-A_I B_I)\left[K_{56I} - K_{34I} + B_I K_{12R}\right] - 2B_I\left[K_{56R} - K_{34R} - B_I K_{12I}\right]}{(1-A_I B_I)^2 + (2B_I)^2} \tag{III-44b}$$

$$C_{22R} = K_{12R} - U_{22R} + A_I U_{22I} \tag{III-44c}$$

$$C_{22I} = K_{12I} - U_{22I} - A_I U_{22R} \tag{III-44d}$$

and V_{22} are known.

3. Lower Surface of a Blade

At a grid point on the lower surface of a blade, the normal perturbation velocity at grid point F_{22} is also known. The computational molecule used is shown in figure 5.

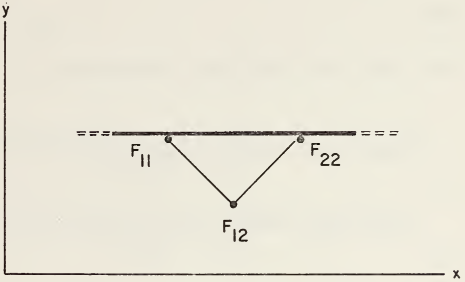


Figure 5. Computational molecule used for a grid point on the lower surface of a blade.

For this grid point, the finite difference equations become,

$$U_{22R} + C_{22R} - A_I U_{22I} = K_{12R} \qquad \text{(III-45a)}$$

$$U_{22I} + C_{22I} + A_I U_{22R} = K_{12I} \qquad \text{(III-45b)}$$

$$U_{22R} - B_I (U_{22I} - C_{22I}) = K_{56R} + K_{34R} \qquad \text{(III-45c)}$$

$$U_{22I} + B_I (U_{22R} - C_{22R}) = K_{56I} + K_{34I} \qquad \text{(III-45d)}$$

where K_{12} is the same as Eqs. (39a) and (39b), but

$$K_{56R} = U_{12R} - V_{12R} + B_I(U_{12I} - C_{12I}) \quad \text{(III-46a)}$$

$$K_{56I} = U_{12I} - V_{12I} - B_I(U_{12R} - C_{12R}) \quad \text{(III-46b)}$$

$$K_{34R} = V_{22R} \quad \text{(III-46c)}$$

$$K_{34I} = V_{22I} \quad \text{(III-46d)}$$

Solving for the perturbation quantities at grid point 22 yields,

$$U_{22R} = \frac{(1-A_I B_I)\left[K_{56R} + K_{34R} - B_I K_{12I}\right] + 2B_I\left[K_{56I} + K_{34I} + B_I K_{12R}\right]}{(1-A_I B_I)^2 + (2B_I)^2} \quad \text{(III-47a)}$$

$$U_{22I} = \frac{(1-A_I B_I)\left[K_{56I} + K_{34I} + B_I K_{12R}\right] - 2B_I\left[K_{56R} + K_{34R} - B_I K_{12I}\right]}{(1-A_I B_I)^2 + (2B_I)^2} \quad \text{(III-47b)}$$

$$C_{22R} = K_{12R} - U_{22R} + A_I U_{22I} \quad \text{(III-47c)}$$

$$C_{22I} = K_{12I} - U_{22I} - A_I U_{22R} \quad \text{(III-47d)}$$

and V_{22} are known.

4. Initial Left Running Mach Line

To obtain the equations for the initial left running Mach line (ξ = constant), the computational molecule of figure 5 is used. However, it is assumed that grid points F_{11} and F_{21} are just in the freestream and the limiting condition $x \to 0$ is taken. For $x \to 0$, A_I and $B_I \to 0$. Thus Eqs.

(38) reduce to,

$$U_{22} + V_{22} = 0 \qquad \text{(III-48a)}$$

$$U_{22} + C_{22} = 0 \qquad \text{(III-48b)}$$

or,

$$U = - V = - C \qquad \text{(III- 49)}$$

Now Eq. (35a) becomes

$$\left(\frac{dU}{dx}\right)_{\xi} = - ik\frac{M^2}{M^2-1}U \qquad \text{(III- 50)}$$

which can be integrated along ξ = constant. Or,

$$U = U\Big|_{x=0}\exp[-ik\frac{M^2}{M^2-1}x] \qquad \text{(III- 51)}$$

$U\Big|_{x=0}$ is known from Eqs. (49) and (37) Thus,

$$U_{22R} = - V_{22R}(0)\cos(k\frac{M^2}{M^2-1}x) - V_{22I}(0)\sin(k\frac{M^2}{M^2-1}x)$$

(III-52a)

$$U_{22I} = V_{22R}(0)\sin(k\frac{M^2}{M^2-1}x) - V_{22I}(0)\cos(k\frac{M^2}{M^2-1}x)$$

(III-52b)

with the appropriate form of Eq. (37) used for $V_{22}(0)$. Then,

$$V_{22R} = - U_{22R} \qquad \text{(III-52c)}$$

$$V_{22I} = - U_{22I} \qquad \text{(III-52d)}$$

$$C_{22R} = - U_{22R} \qquad \text{(III-52e)}$$

$$C_{22I} = - U_{22I} \tag{III-52f}$$

Eqs. (52) are also used to compute the perturbation quantities at the leading edge of the upper surface of the first blade.

5. Initial Right Running Mach Line

The developement for the perturbation quantities along the initial right running Mach line (η = constant) is analagous to the development for the left. Here, F_{12} and F_{11} of figure 2 are assumed to be just in the freestream. Eqs. (38) reduce to,

$$U_{22} - V_{22} = 0 \tag{III-53a}$$

$$U_{22} + C_{22} = 0 \tag{III-53b}$$

or,

$$U = V = - C \tag{III- 54}$$

Then Eq. (35b) becomes,

$$\left(\frac{dU}{dx}\right)_{\eta} = - ik\frac{M^2}{M^2-1}x \tag{III- 55}$$

Integrating along η = constant yields,

$$U = U\Big|_{x=0} \exp[-ik\frac{M^2}{M^2-1}x] \tag{III- 56}$$

or,

$$U_{22R} = V_{22R}(0)\cos(k\frac{M^2}{M^2-1}x) + V_{22I}(0)\sin(k\frac{M^2}{M^2-1}x) \tag{III-57a}$$

$$U_{22I} = - V_{22R}(0)\sin(k\frac{M^2}{M^2-1}x) + V_{22I}(0)\cos(k\frac{M^2}{M^2-1}x) \tag{III-57b}$$

Again, the appropriate form of Eq. (37) is used to define $V_{22}(0)$. Then,

$$V_{22R} = U_{22R} \tag{III-57c}$$

$$V_{22I} = U_{22I} \tag{III-57d}$$

$$C_{22R} = - U_{22R} \tag{III-57e}$$

$$C_{22I} = - U_{22I} \tag{III-57f}$$

Eqs. (57) are used to compute the perturbation quantities at the first grid point on the lower surface of the first blade. In the computer program, lower surface points at grid point F_{22} are subscripted 33 or 55 for pitch and plunge respectively.

6. First Grid Point on a New Blade

For the first grid point encountered on any blade other than the first, the computational molecule of figure 6 is used. The new blade is assumed to protrude an infinitisimal distance past grid point F_{22} which is labeled F_{22U} and F_{22L} in the figure. Then, the equations developed for grid points on the upper and lower surface of a blade are used to compute the perturbation quantities at F_{22U} and

F_{22L} respectively.

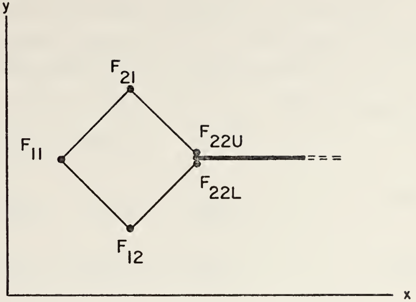


Figure 6. Computational molecule used for the first grid point on a new blade.

(7) Wake Grid Points

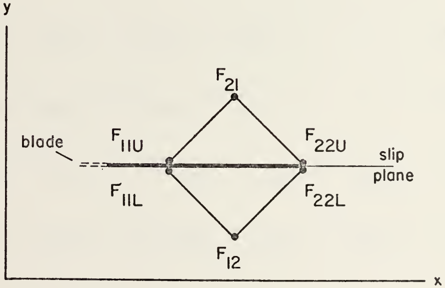


Figure 7. Computational molecule used for a grid point in the wake.

It is assumed that the first wake grid point is an infinitismal distance aft of the blade. To calculate the

flow field quantities in the wake, the computational molecule is divided into two triangular molecules as shown. Then, the equations developed for a grid point on the upper surface of a blade apply to F_{22U} while those for the lower surface of a blade apply to F_{22L}. The computational molecule is shown in figure 7.

$$U_{22RU} + C_{22RU} - A_I U_{22IU} = K_{12RU} \qquad \text{(III-58a)}$$

$$U_{22IU} + C_{22IU} + A_I U_{22RU} = K_{12IU} \qquad \text{(III-58b)}$$

$$U_{22RU} + V_{22RU} + B_I (C_{22IU} - U_{22IU}) = K_{56R} \qquad \text{(III-58c)}$$

$$U_{22IU} + V_{22IU} - B_I (C_{22RU} - U_{22RU}) = K_{56I} \qquad \text{(III-58d)}$$

and,

$$U_{22RL} + C_{22RL} - A_I U_{22IL} = K_{12RL} \qquad \text{(III-59a)}$$

$$U_{22IL} + C_{22IL} + A_I U_{22RL} = K_{12IL} \qquad \text{(III-59b)}$$

$$U_{22RL} - V_{22RL} + B_I (C_{22IL} - U_{22IL}) = K_{34R} \qquad \text{(III-59c)}$$

$$U_{22IL} - V_{22IL} - B_I (C_{22RL} - U_{22RL}) = K_{34I} \qquad \text{(III-59d)}$$

where,

$$K_{12RU} = U_{11RU} + C_{11RU} + A_I U_{11IU} \qquad \text{(III-60a)}$$

$$K_{12IU} = U_{11IU} + C_{11IU} - A_I U_{11RU} \qquad \text{(III-60b)}$$

$$K_{12RL} = U_{11RL} + C_{11RL} + A_I U_{11IL} \qquad \text{(III-60c)}$$

$$K_{12IL} = U_{11IL} + C_{11IL} - A_I U_{11RL} \qquad \text{(III-60d)}$$

$$K_{34R} = U_{12R} - V_{12R} + B_I (U_{12I} - C_{12I}) \qquad \text{(III-60e)}$$

$$K_{34I} = U_{12I} - V_{12I} - B_I (U_{12R} - C_{12R}) \qquad \text{(III-60f)}$$

$$K_{56R} = U_{21R} + V_{21R} + B_I (U_{21I} - C_{21I}) \qquad \text{(III-60g)}$$

$$K_{56I} = U_{21I} + V_{21I} - B_I (U_{21R} - C_{21R}) \qquad \text{(III-60h)}$$

Eqs. (58) - (60) form a system of eight equations with twelve unknowns. However, in the linearized problem, the wake cannot support a pressure difference, and the normal velocity across the wake must be a constant. As a result of these two conditions, to first order effect, there will be no discontinuities reflected from the wake. Therefore, the relations,

$$V_{22RU} = V_{22RL} = V_{22R} \qquad \text{(III-61a)}$$

$$V_{22IU} = V_{22IL} = V_{22I} \qquad \text{(III-61b)}$$

$$C_{22RU} = C_{22RL} = C_{22R} \qquad \text{(III-61c)}$$

$$C_{22IU} = C_{22IL} = C_{22I} \qquad \text{(III-61d)}$$

are used to reduce Eqs (58) and (59) to a determinant eight by eight system of equations which are:

$$U_{22RU} + C_{22R} - A_I U_{22IU} = K_{12RU} \qquad \text{(III-62a)}$$

$$U_{22IU} + C_{22I} + A_I U_{22RU} = K_{12IU} \qquad \text{(III-62b)}$$

$$U_{22RU} + V_{22R} + B_I (C_{22I} - U_{22IU}) = K_{56R} \qquad \text{(III-62c)}$$

$$U_{22RU} + V_{22I} - B_I (C_{22R} - U_{22RU}) = K_{56I} \qquad \text{(III-62d)}$$

and

$$U_{22RL} + C_{22R} - A_I U_{22IL} = K_{12RL} \qquad \text{(III-63a)}$$

$$U_{22IL} + C_{22I} + A_I U_{22RL} = K_{12IL} \qquad \text{(III-63b)}$$

$$U_{22RL} + V_{22R} + B_I (C_{22I} - U_{22IL}) = K_{34R} \qquad \text{(III-63c)}$$

$$U_{22IL} + V_{22I} - B_I (C_{22R} - U_{22RL}) = K_{34I} \qquad \text{(III-63d)}$$

Eqs. (60) remain unchanged.

C. THE PRESSURE COEFFICIENT

The non-dimensional pressure distribution along the surface of a blade is determined using the linearized pressure difference of Eq. (II-48) and Teipel's sonic velocity perturbation amplitude function, Eq. (21c). Rewriting Eq. (II-48) as,

$$p - p_o = \frac{2}{\gamma-1} \rho_o c_o^2 \left(\frac{\bar{c} - c_o}{c_o} \right) \qquad \text{(III- 64)}$$

and then dividing by the freestream dynamic pressure yields

$$\frac{p - p_o}{\frac{1}{2}\gamma p M^2} = 2C(x,y)e^{ikt} \qquad \text{(III- 65)}$$

Hence, the complex pressure difference across a blade at grid point P_{22} resulting from oscillation in pitch or plunge is,

$$P(x,y,t) = \frac{p_L - p_U}{\frac{1}{2}\gamma p M^2} = 2(C_{22L} - C_{22U})e^{ikt} \qquad \text{(III-66a)}$$

This expression for the pressure difference across a blade appears to be independent of interblade phase angle δ. However, the interblade phase angle was used for the computation of the normal velocity through the flow tangency equations, and they were used in the determination of C. Therefore, to compute the pressure distribution along the n^{th} blade when it is in a specific position requires that the position be specified in terms of the position of the first blade. In general, the complex pressure is

$$P(x,y,t) = (P_R + iP_I)e^{ikt} \qquad \text{(III- 67)}$$

with P_R and P_I defined by Eq. (66). The position of the n^{th} blade is

$$(kt)_n = (kt) + (n-1)\delta \qquad \text{(III- 68)}$$

Then expanding Eq. (67),

$$P(x,y,t) = [P_R\cos(kt) - P_I\sin(kt)]$$
$$i[P_R\sin(kt) + P_I\cos(kt)] \qquad \text{(III- 69)}$$

Figure 8 shows a complete oscillation cycle for a blade undergoing simple hamonic motion in pitch. The extension to plunge is obvious, and the following discussion is applicable there also.

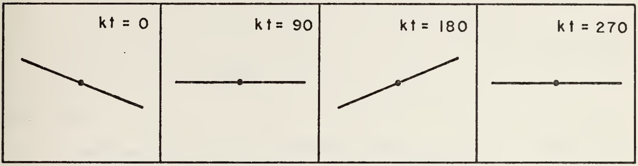


Figure 8. Blade oscillation cycle.

The two conditions of primary interest here are when the n^{th} blade is in the initial position

$$(kt)_n = 0 \qquad \text{(III- 70)}$$

or in the mean position, pitching leading edge up,

$$(kt)_n = 270 \qquad \text{(III- 71)}$$

Then solving for the position of the first blade when the n^{th} blade is in the initial position gives,

$$(kt) = -(n-1)\delta \qquad \text{(III- 72)}$$

and in the mean position,

$$(kt) = 270 - (n-1)\delta \qquad \text{(III- 73)}$$

Except for flutter analysis, the real component of the pressure is normally of primary interest, and can be found by solving Eq. (69) with (kt) expressed as a function of

$(kt)_n$. For ease of showing the relation of the real component of pressure for the two blade positions above, consider the special case of $\delta = 0$. Then,

$$\mathrm{Re}\{P(x,y,t)\}_{kt=0} = P_R \qquad \text{(III-74a)}$$

$$\mathrm{Re}\{P(x,y,t)\}_{kt=270} = P_I \qquad \text{(III-74b)}$$

and thus,

$$\mathrm{Re}\{P(x,y,t)\}_{kt=0} = \mathrm{Im}\{P(x,y,t)\}_{kt=270} \qquad \text{(III-75a)}$$

$$\mathrm{Re}\{P(x,y,t)\}_{kt=270} = \mathrm{Im}\{P(x,y,t)\}_{kt=0} \qquad \text{(III-75b)}$$

Having obtained the pressure distribution over the blades, the dimensionless lift per unit span and the moment per unit span square are found from,

$$L(x,y,t) = = \int_0^1 P(x,y,t)\,dx \qquad \text{(III-76a)}$$

and

$$M(x,y,t) = = \int_0^1 (x-x_0)\,P(x,y,t)\,dx \qquad \text{(III-76a)}$$

where L(x,y,t) and M(x,y,t) are complex quantities which include the effects of blade oscillation in pitch and plunge. Lift is positive upward and moments are positive clockwise (see figure 2).

IV. FLUTTER ANALYSIS

A. EQUATIONS OF MOTION FOR A SINGLE BLADE

Lane [Ref. 14] showed that cascade flutter could be analyzed by considering only a single oscillating blade. Following Fung [Ref. 8], the equations of motion for a flat plate airfoil immersed in an inviscid flow are derived from a force balance of the inertia, elastic and aerodynamic forces. For bending motion, h is measured positive down from the elastic axis. The torsional motion is positive clockwise about the elastic axis (refer to figure 2). It is assumed that a pair of springs, with spring constants K_α and K_h, oppose the blade motion.

The total inertia force per unit span is,

$$F_i = -(M_b \ddot{h} + S_b \ddot{\alpha}) \qquad \text{(IV- 1)}$$

and this force exerts a moment about the elastic axis x_o of,

$$M_i = -(I_\alpha \ddot{\alpha} + S_b \ddot{h}) \qquad \text{(IV- 2)}$$

where

M_b - total blade mass per unit span

S_b - wing static moment (about x_o)

I_α - wing mass moment of inertia (about x_o)

The springs produce restoring forces and moments of

$$F_r = -hK_h \qquad \text{(IV-3a)}$$

$$M_r = -\alpha K_\alpha \qquad \text{(IV-3b)}$$

Then the equations of motion for the blade are,

$$\text{Lift/span} = M_b\ddot{h} + S_b\ddot{\alpha} + hK_h \qquad \text{(IV-4a)}$$

$$\text{Moment/span} = S_b\ddot{h} + I_\alpha\ddot{\alpha} + \alpha K_\alpha \qquad \text{(IV-4b)}$$

The assumption of simple harmonic motion is expressed as,

$$h = h_o e^{i\omega T} \qquad \text{(IV-5a)}$$

$$\alpha = \alpha_o e^{i\omega T} \qquad \text{(IV-5b)}$$

where h_o and α_o are complex amplitudes. The blades oscillate with a constant interblade phase angle δ. A phase difference exists between twist and flexure also, but cannot be determined until a flutter solution is obtained. Then, the ratio of blade deflection amplitudes will give this phase difference.

With Eqs. (5), the equations of motion for the blade become,

$$\text{Lift/span} = -\omega^2 M_b h_o - \omega^2 S_b \alpha_o + h_o K_h \qquad \text{(IV-6a)}$$

$$\text{Moment/span} = -\omega^2 S_b h_o - \omega^2 I_\alpha \alpha_o + \alpha_o K_\alpha \qquad \text{(IV-6b)}$$

The natural frequencies of twist and flexure are found by assuming that each mode can exist separately, in the absence of exciting forces. Then,

$$\omega_h = \sqrt{K_h/M_b} \qquad \text{(IV-7a)}$$

$$\omega_\alpha = \sqrt{K_\alpha/I_\alpha} \qquad \text{(IV-7b)}$$

or,

$$K_h = M\omega_h^2 \qquad \text{(IV-8a)}$$

$$K_\alpha = I_\alpha\omega_\alpha^2 \qquad \text{(IV-8b)}$$

and therefore,

$$\text{Lift/span} = [-h_o(\omega^2 M_b - M_b\omega_h^2) - \alpha_o\omega^2 S_b]e^{i\omega T} \qquad \text{(IV-9a)}$$

$$\text{Moment/span} = [-h_o\omega^2 S_b - \alpha_o(\omega^2 I_\alpha - I_\alpha\omega_\alpha^2)]e^{i\omega T} \qquad \text{(IV-9b)}$$

B. TWO DEGREE OF FREEDOM FLUTTER EQUATIONS

The method used to investigate flutter in this report follows the procedure and notation used by Garrick and Rubinow [Ref. 9], who derived the lift force and moment equations,

$$L_{G/R} = -4\rho b^3\omega^2 e^{i\omega T}\left[\frac{h_o}{b}(L_1 + iL_2) + \alpha_o(L_3 + iL_4)\right] \qquad \text{(IV-10a)}$$

$$M_{G/R} = -4\rho b^4\omega^2 e^{i\omega T}\left[\frac{h_o}{b}(M_1 + iM_2) + \alpha_o(M_3 + iM_4)\right] \qquad \text{(IV-10b)}$$

where $L_{G/R}$ was defined as positive down. Equating the magnitudes of the lift from Eqs. (9a) and (10a), one obtains,

$$\frac{h_o}{b}(L_1 + iL_2 - \mu + \Omega_h X) + \alpha_o(L_3 + iL_4 - \mu x_\alpha) = 0$$

(IV-11)

and by equating the amplitude of the moments,

$$\frac{h_o}{b}(M_1 + iM_2 - \mu x_\alpha) + \alpha_o(M_3 + iM_4 - \mu r_\alpha^2 + \Omega_h X) = 0 .$$

(IV- 12)

where the non-dimensional terms are defined as,

$(L_1 + iL_2), (M_1 + iM_2)$ - real and imaginary component of lift and moment due to plunge.

$(L_3 + iL_4), (M_3 + iM_4)$ - real and imaginary component of lift and moment due to pitch.

μ - wing density parameter $\left(\frac{M_b}{4\rho b^2}\right)$

Ω_h - ratio of natural frequencies of flexure and torsion $\mu\left(\frac{\omega_h}{\omega_\alpha}\right)^2$

Ω_α - the quantity μr_α^2.

x_α - location of the wing center of gravity measured from the elastic axis $\left(\frac{S_b}{M_b b}\right)$

r_α - radius of gyration of the wing $\left(\frac{I_\alpha}{M_b b^2}\right)$

X - ratio of natural frequency in pitch to the circular frequency $\left(\frac{\omega_\alpha}{\omega}\right)^2$

Experimental results indicate the $\omega_\alpha > \omega_h$ in modern turbomachines [Ref. 18].

Eqs. (11) and (12) form an eigenvalue problem in $\left(\frac{h_o}{b}\right)$

and (α_0) which has a non-trivial solution if and only if the determinant of the coefficient matrix vanishes. Equating the determinant to zero yields a complex function in the unknown X, each component of which must be identically equal to zero. These equations are,

$$\Omega_h \Omega_\alpha X^2 + [\Omega_\alpha (L_1 - \mu) + \Omega_h (M_3 - \Omega_\alpha)]X + C_R = 0 \quad \text{(IV-13a)}$$

$$(\Omega_\alpha L_2 + \Omega_h M_4)X + C_I = 0 \quad \text{(IV-13b)}$$

$$C_R = \mu[x_\alpha(M_1 + L_3) - (M_3 - \Omega_\alpha) - L_1 r_\alpha^2 - \mu x_\alpha^2] + D_R \quad \text{(IV-13c)}$$

$$C_I = \mu[x_\alpha(M_2 + L_4) - M_4 - L_2 r_\alpha^2] + D_I \quad \text{(IV-13d)}$$

$$D_R = L_1 M_3 - L_3 M_1 - L_2 M_4 + L_4 M_2 \quad \text{(IV-13e)}$$

$$D_I = L_1 M_4 - L_4 M_1 + L_2 M_3 - L_3 M_2 \quad \text{(IV-13f)}$$

For a fixed cascade geometry and interblade phase angle, Eqs. (13) are all transcendental functions of the reduced frequency k, or (1/k), more accurately. The solution of Eqs. (13a) and (13b) is found using the Theodorsen method. $\sqrt{\text{Re}\{X\}}$ and $\sqrt{\text{Im}\{X\}}$ are tabulated and plotted versus (1/k). The flutter point is defined by the intersection of the real and imaginary curves

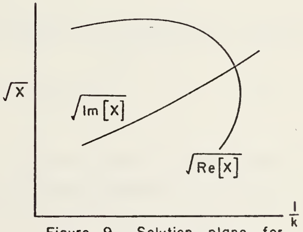


Figure 9. Solution plane for the Theodorsen method.

of $\sqrt{X}$. The approximate shape of these curves is shown in figure 9. Since (1/k) appears transcendentally in every term of Eqs. (13), the curve of $\sqrt{Re\{X\}}$ is not exactly parabolic, and for $\sqrt{Im\{X\}}$, not actually linear.

At the flutter point, the non-dimensional flutter frequency is,

$$\frac{\omega_F}{\omega_\alpha} = \frac{1}{\sqrt{X}} \qquad \text{(IV-14a)}$$

and the non-dimensional flutter speed is,

$$\frac{U_F}{\omega \hat{c}} = \frac{1}{k}\frac{1}{\sqrt{X}} \qquad \text{(IV-14b)}$$

k is defined by Eq. (II-39).

C. SINGLE DEGREE OF FREEDOM FLUTTER

For single degree of freedom oscillations in pitch, the equation of motion for the blade is,

$$\text{Moment/span} = I_\alpha \ddot{\alpha} + \alpha K_\alpha \qquad \text{(IV- 15)}$$

which with the assumption of simple harmonic motion becomes,

$$\text{Moment/span} = [- \omega^2 I_\alpha \alpha_0 + K_\alpha \alpha_0]e^{i\omega T} \qquad \text{(IV- 16)}$$

The mechanical damping of the system is introduced by assuming the spring opposing the displacement in pitch to have a spring constant of the form $K_\alpha(1 + ig_\alpha)$ [Ref. 8]. The natural frequency in pitch is determined as before. Then, equating the amplitude of this expression for the

moment with that of Eq. (10a) with $(\frac{h_o}{b}) = 0$, one obtains,

$$\Omega_{\alpha}(X - 1) + M_{3} = 0 \qquad \text{(IV-17a)}$$

$$M_{4} + g_{\alpha}\Omega_{\alpha} X = 0 \qquad \text{(IV-17b)}$$

The solution method is the same as for the two degree of freedom problem, and the non-dimensional flutter frequency and speed are found from Eqs. (14). A similar development is required to obtain the equations for single degree of freedom plunge. However, based on the results of this, and other works, plunge oscillations are always stable.

For both single degree of freedom and coupled flexure-torsion motion, the flutter calculations for the nth blade must be repeated for varying interblade phase angles until the minimum flutter speed is found. This speed is termed the critical flutter speed. For all speeds below it, cascade operation will be stable.

An alternate method for determining the onset of flutter is to compute the real component (the in phase component) of the work per cycle of the blade on the freestream,

$$\text{Work/cycle} = \mathrm{Re}\left\{\int_{0}^{2\pi/\omega} (M_{\alpha}\dot{\alpha}dt + L_{h}\dot{h}dt)\right\} \qquad \text{(IV- 18)}$$

As long as the blade does positive work on the freestream, the cascade will be stable. However, if the aerodynamic damping decreases to the point that the airstream does work on the blade, flutter will occur. For single degree of freedom pitch oscillations, with zero structural damping, the sign of the imaginary component of the moment determines whether the work/cycle is positive or negative, and can therefore be used to indicate the onset of flutter

[Ref. 4,18].

D. DEFINITION OF TERMS

The complex pressure difference at each grid point along a blade was shown by Eq. (III-66) to be a direct function of the sonic velocity perturbation amplitude. Eqs. (III-76) then gave the non-dimensional lift force and moment per unit span. These integrals were numerically computed using a trapezoidal integration routine. If F represents the integrated value of (m+1) discret values of f(i), then,

$$F = f(x)\,dx = \{\frac{1}{2}[f(1) + f(m+1)] + \sum_{i} f(i)\}\Delta x \qquad \text{(IV- 19)}$$

To relate the non-dimensional expressions for lift force and moment as derived by Garrick and Rubinow to those computed using the Teipel amplitude functions, the following quantities are defined:

$$P(x,y,t) = [\frac{h_o}{\hat{c}}(P_h) + \alpha_o(P_\alpha)]e^{i\omega T} \qquad \text{(IV-20a)}$$

Then the dimensionsal lift force and moment per unit span are,

$$\text{Lift} = \frac{1}{2}\rho u_0^2 \hat{c} L(x,y,t) = \frac{1}{2}\rho u_0^2 \hat{c}[\frac{h_o}{\hat{c}}(L_h) + \alpha_0(L_\alpha)\}e^{ikt} \qquad \text{(IV-20b)}$$

$$\text{Moment} = \frac{1}{2}\rho u_0^2 \hat{c}^2 M(x,y,t) = \frac{1}{2}\rho u_0^2 \hat{c}^2[\frac{h_o}{\hat{c}}(M_h) + \alpha_0(M_\alpha)]e^{ikt} \qquad \text{(IV-20c)}$$

L_h, M_h, L_α and M_α are all complex quantities. The subscript

indicates the perturbation source. Garrick and Rubinow defined the lift force as positive down. Therefore, equating Eqs. (20b,c) to the negative of the corresponding components of Eqs. (10), one obtains,

$$\frac{1}{k^2}\left[\frac{h_o}{\bar{c}}(L_h) + (L_\alpha)\right] = \left[\frac{h_o}{b}(L_1 + iL_2) + \alpha_o L_3 + iL_4)\right] \qquad \text{(IV-21a)}$$

$$\frac{2}{k^2}\left[\frac{h_o}{\bar{c}}(M_h) + (M_\alpha)\right] = \left[\frac{h_o}{b}(M_1 + iM_2) + \alpha_o(M_3 + iM_4)\right] \qquad \text{(IV-21b)}$$

using the definition of reduced frequency from Eq. (II-39). Hence,

$$L_1 = \frac{1}{2k^2}\,\mathrm{Re}\{L_h\} \qquad \text{(IV-22a)}$$

$$L_2 = \frac{1}{2k^2}\,\mathrm{Im}\{L_h\} \qquad \text{(IV-22b)}$$

$$L_3 = \frac{1}{k^2}\mathrm{Re}\{L_\alpha\} \qquad \text{(IV-22c)}$$

$$L_4 = \frac{1}{k^2}\,\mathrm{Im}\{L_\alpha\} \qquad \text{(IV-22d)}$$

$$M_1 = \frac{1}{k^2}\,\mathrm{Re}\{M_h\} \qquad \text{(IV-22e)}$$

$$M_2 = \frac{1}{k^2}\,\mathrm{Im}\{M_h\} \qquad \text{(IV-22f)}$$

$$M_3 = \frac{2}{k^2}\,\mathrm{Re}\{M_\alpha\} \qquad \text{(IV-22g)}$$

$$M_4 = \frac{2}{k^2} \operatorname{Im}\{M_\alpha\} \tag{IV-22h}$$

The computer program, which is listed after the appendices and is explained in Section V, computes the non-dimensional lift force and moment defined by Eqs. (20b,c) for each blade of the cascade. These values are then modified by Eqs. (22) so that the flutter analysis can be made for the n^{th} blade.

V. COMPUTER PROGRAM

A. INTRODUCTION

The finite difference equations derived in Section III-B and the flutter equations of Section IV were programmed in FORTRAN IV and run on an IBM 360/67 computer. The computer program listing for the two degree of freedom case of pitch and plunge is given following Appendix B.

The computational procedure is the same for pitch or plunge, the only difference between the two being the form of the flow tangency equation. The two degree of freedom solution is simply a superposition of the individual cases of pitch and plunge.

It should be noted that the subscripts of the four grid points of the computational molecule shown in figure 3 are distinct from the notation used for subscripting the perturbation quantities in the computer program. There, the subscripts indicate whether or not a perturbation quantity at a grid point is above or below a surface, and if the perturbation is due to oscillation in pitch or plunge.

In the computer program, all perturbation quantities have an alpha-numeric subscript (two numbers and a letter) and an index. The index identifies the grid point. The letter in the subscript tells whether the component is real (R) or imaginary (I), while the numbers indicate:

22 (pitch) or 44 (plunge) ..... perturbation quantity is at a grid point which is not on the lower surface of a blade or wake slip plane.

33 (pitch) or 55 (plunge) ..... perturbation quantity is at a grid point located on the lower surface of a blade or slip plane.

Thus, V_{55I} is the complex component of the perturbation in normal velocity, due to plunge, on the lower surface of a blade or slip plane. The subscript of the perturbation quantity solved for in this section merely indicates the real or imaginary component of that quantity at grid point 22.

B. PROGRAM CONSTRAINTS

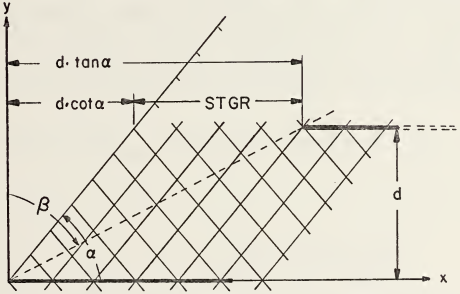


Figure 10. The compatible stagger angle.

The computer program is quite general in its ability to solve a given cascade flow problem. The one geometric constraint that must be satisfied by the user is Eq. (II-1), subsonic leading edge locus of the cascade blading. The computer program requires that the leading edge of each

blade be exactly on a grid point. Therefore, the input stagger angle is modified so that STGR, shown in figure 10, is a non-zero integer multiple of DSTSTR aft of the initial Mach wave. Then the 'compatible' stagger angle is determined and given in the output. The amount by which this procedure alters the input cascade geometry is a function of the gridfineness, Mach number, solidity, and distance between blades. See also V-G.

The only other restrictions are on vector size. Internal checks are made at the beginning of a run to insure that the program dimensions are large enough for the input geometry and Mach number. If the storage area is exceeded, program termination occurs. Of course, the program is only valid for $M > 1$, and in the range where linearized theory is valid.

The vector dimension requirements for a given cascade and Mach number can be determined prior to running the program. The vectors used for storage of all perturbation quantities and grid point locations along a right-running Mach line must be dimensioned an integer value greater than,

$$\frac{1}{2}\epsilon \tan(\bar{\alpha})\{(N_{bld}-1)[\tan(\beta) - \cot(\bar{\alpha})] + \frac{1}{d}\} + 1 \qquad (V-1)$$

where,

ϵ . . . . grid fineness ratio

N_{bld} . . . . number of blades

d . . . . distance between blades

and $\bar{\alpha}$ and β have been defined previously.

The program marches down successive right-running Mach lines from one grid point to the next. The location of a grid point, along the current Mach line, at which the

perturbation quantities have just been computed is IHAVEP. The next grid point is labeled IHAVEP+1. In the subroutines, these same two grid points are labeled J and I respectively. Then IHAVEP+1 = I is the index of the storage location within a vector array of a perturbation quantity or grid point location at F_{22} of figure 3. The left-running Mach line which passes through grid points F_{11} and F_{21} is a line of constant IHAVEP = J, and the next left-running Mach line is a line of constant IHAVEP+1 = I.

The last grid point in the flow field is the first wake grid point aft of the last blade in the cascade. Along the left-running Mach line which passes through this grid point, IHAVEP = LIMIT. When the computation reaches this Mach line, the program switch variables are reset, and the computational procedure jumps to the next right-running Mach line. Thus, perturbation quantity indices are from 1 to LIMIT+1, while IHAVEP goes from 0 to LIMIT along a Mach line.

The perturbation quantity vectors are single dimension arrays in order to reduce required core space. However, as the computational molecule requires perturbation quantities from two different Mach lines, it is necessary to temporarily switch and store the perturbation quantity values as scalars upon entrance to each subroutine before solving for 'new' values at F_{22}. Figure 11 shows the storage locations of a representative perturbation quantity vector as it is in the program on entrance to and exit from a subroutine.

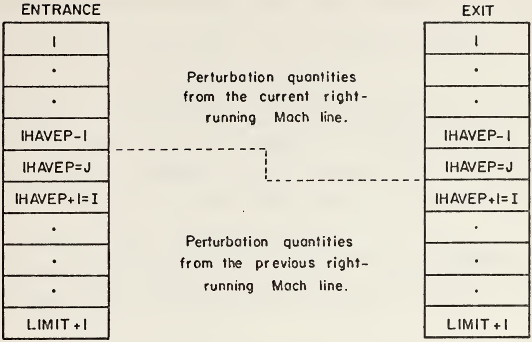


Figure 11. Typical perturbation quantity vector storage on entrance to and exit from a subroutine.

The computational procedure in subroutines GENFPT, NEWBLD, TOP, BOTTOM and WAKE is the same. This procedure is:

Step 1. The incomming perturbation quantities at grid point IHAVEP=J on the current Mach line (F_{21} of figure 3) are stored as the scalar quantities having suffixes NEW (U2RNEW, V4INEW, etc). Stored in the vector on the left at location IHAVEP are the perturbation quantities from the previous Mach line (F_{11}).

Step 2. The constants K12R, K12I, etc., defined in Section III-B, are computed using the perturbation

quantities at grid points F_{11} and F_{21}. Now, perturbation quantities from the previous Mach line, stored at location IHAVEP (vector on the left) , are no longer required.

Step 3. Perturbation quantities from the previous grid point of the current Mach line (U2RNEW, V4INEW, etc.) are stored in the vector at position IHAVEP=J (vector on right), which removes from memory the perturbation quantities from the previous Mach line at grid point IHAVEP.

Step 4. The perturbation quantities at IHAVEP+1 = I are computed and stored as the scalar quantities U2RNEW, V4INEW, etc., and the subroutine is exited.

This procedure is slightly different in subroutines BOTTOM, WAKE and NEWBLD because perturbation quantities for lower surface points are computed. When a lower surface point has been computed by one of these subroutines, the integer variable ICO is set equal to one. This flagged value then causes GENFPT, which will always be the next subroutine called, to use lower surface perturbation quantities in the computational molecule at grid point F_{21} for the calculation of perturbation quantities at grid point F_{22}.

The last grid point on all right-running Mach lines will lie in either the general flow field or the wake of one of the blades. When this grid point is reached, after returning from the appropriate subroutine to MAIN, the perturbation quantities just computed are stored. Then the program jumps to the next right-running Mach line. The program terminates flow field calculations on the first grid point in the wake aft of the last blade.

Subroutine COEF stores the pressure difference across a

blade resulting from pitch or plunge. These values are then used to compute the lift and moment (about the elastic axis, x_o) for each blade. In order to be able to identify these pressure differences with the appropriate blade, the information is stored in a two-dimensional matrix. The first dimension identifies the blade, while the second dimension specifies the maximum number of grid points along the blade. In order to minimize the required core size, it was desirable not to have the first dimension equal the total number of blades in the cascade. Instead, a smaller matrix was used cyclically for all of the blades. However, this required that a restriction be placed on the number of blades that any right-running Mach line can cross.

After the first wake point aft of a blade is reached, COEF2 (alternate entry point for subroutine COEF) is called, and the pressure difference at the end point of the blade (x_{rel} = 1) linearly extrapolated. Then the dimensionless lift force and moment, as given by Eqs. (III-76), are computed using a trapezoidal integration technique. Once the lift and moment have been computed that portion of the matrix storing information relative to that blade may be reused for another blade. The maximum number of blades that a right-running Mach line may cross is ICROSS, which is specified in subroutine INPUT by a data statement. In order not to exceed the storage space allocated for these vectors storing pressure and moment distributions, the right-running Mach line which passes through the leading edge of the $(n + \text{ICROSS} + 1)^{th}$ blade must pass aft of the first wake grid point of the n^{th} blade. Also, there must be less than MAXPTS grid points along a blade, including the extrapolated

point at $x_{rel} = 1$. ICROSS and MAXPTS are the dimensions of the vectors storing the pressure and moment distributions.

Letting $| \ |$ denote the chopped integer value of a real number, the restriction on blades crossed is satisfied if

$$\mathrm{ICROSS}\{|d[\tan(\beta)-\cot(\bar{\alpha})]/\Delta x| + \epsilon\} > \left|\frac{1}{\Delta x} + 2\right| \tag{V- 2}$$

The right hand side of Eq. (2) gives the total number of grid points along the blade (which must be less than MAXPTS). x is the step size along a streamline (DSTSTR), and is equal to

$$\Delta x = 2d[\cot(\bar{\alpha})]/\epsilon \tag{V- 3}$$

All terms were previously defined in Eq. (1).

One final dimension that must not be exceeded is the total number of blades, which can not be greater than MAXBLD. MAXPTS and MAXBLD are specified by a data statement in INPUT in the same manner as ICROSS. Integrated lift forces and moments are stored by blade number. From figure 11, it can be seen that apparent blades are 'seen' by the computer. Depending on the input parameters (low Mach number and high solidity), the computer may 'see' several apparent blades. Therefore, MAXBLD should be at least two or three greater than the number of blades in the cascade. The required core size is insensitive to the magnitude of MAXBLD.

Thus , the constraints and restrictions on program use are given by Eq. (II-1) and Eqs. (V-1) - (V-3) and the comment on the maximum number of blades above. All of the constraints are checked internally by the program in INPUT, and an appropriate error termination message given if one or

more of the constraints is not met.

C. PROGRAM FLOW AND CONTROL

With the exception of the main program and subroutine COEF, all subroutines operate in a straightforward manner to solve the finite difference equations developed in Section III-B. Hence, they will be only briefly discussed.

MAIN controls the program flow as consecutive right-running Mach line are traversed. Integer switch variables, listed in Appendix B, are used in the the five primary logic tests. These are, in the order encountered,

(1) If (LCOUNT = 0) grid point is on the initial right-running Mach line
(2) If (IHAVEP = LCOUNT) grid point is on a blade or in a wake
(3) If (IHAVEP = 0) grid point is on the initial left-running Mach line
(4) If (IHAVEP = NUMPI*NSTPTS) and (JLINE = MAXI) grid point is on the leading edge of a new blade

If (2) was satisfied, then,

(5) If (IHAVEP > LIMW) grid point is in the wake

To clarify program operation and the use of these switch variables, the sample case shown in figure 12 is discussed.

For M = 1.3 and the geometry given, the quantities listed in figure 12 are computed prior to beginning any flow field calculations. ' is the compatible stagger angle computed by the program. Subroutine INITAL calculates the

perturbation quantities at the leading edge of the first blade and initializes IHAVEP = 1, JLINE = 0 and LCOUNT = 0. MAIN then computes NEWDST and LIMIT as shown in the figure, and initializes

LIMW = NEWDST = 10
MAXI = NGRDFN + NSTPTS = 10
NUM = 0
NUMOLD = 0
NUMPI = 1
OLDL = 0
ICO = 0

Now the loop which calculates the entire flow field is entered.

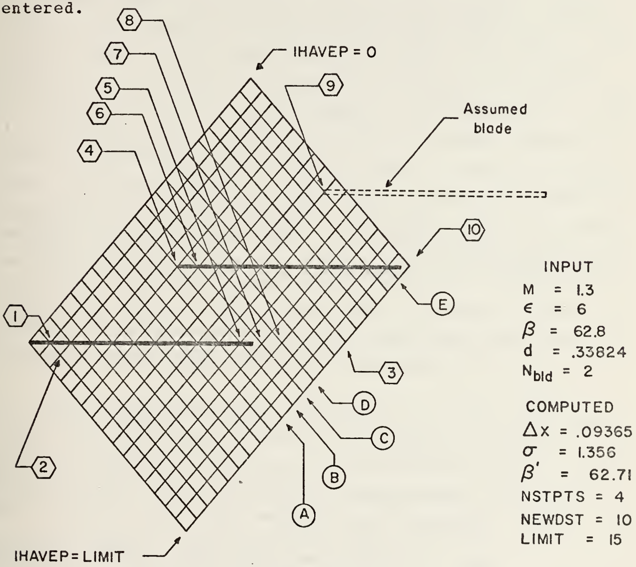


Figure 12. Sample Cascade Problem

IHAVEP = 0 means that the grid point is on the initial left-running Mach line, while IHAVEP ≠ 0 , LCOUNT = 0

indicates that the grid point is on the initial right-running Mach line. In either case, subroutine MACHLN is called to compute the the perturbation quantities associated with each grid point. For computations everywhere except along the initialright-running Mach line, the temporary shuffling of perturbation quantities must be done prior to computing the values at the new grid point.

After computing down the initial Mach line to IHAVEP = LIMIT, the program arrives at the first grid point on the second right-running Mach line with IHAVEP = 0 and LCOUNT = JLINE = 1. All other integer switch variables are the same as before. At the second grid point on this Mach line (point 1 in figure 11), IHAVEP = LCOUNT = 1, which results in a call to TOP/BOTTOM (since IHAVEP < LIMW). Prior to calling TOP, ICO is set equal to 1 so that the next time GENFPT is called (at 2) the perturbation quantities from the lower surface of the blade will be used in the computational molecule for grid point F_{21}. After return from BOTTOM, COEF1 (alternate entry to subroutine COEF) is called to store the pressure difference just computed across the first blade at this grid point. Then IHAVEP is incremented by 1.

For the remaining grid points along this Mach line, since all of the logical tests will be false, GENFPT will be called by default. Again, ICO = 1 indicates that a lower surface point is to be used for perturbation quantities at grid point F_{21} in the computational molecule. Prior to exiting GENFPT, ICO is reset equal to zero. For the remainder of the grid points on the Mach line. GENFPT will be called until IHAVEP = LIMIT. The perturbation quantities computed at each grid point are stored in vector locations IHAVEP+1. When IHAVEP = LIMIT, the integer switches are reset to

IHAVEP = 0

LCOUNT = LCOUNT+1 = 2
JLINE = JLINE+1 = 2
ICO = 0

ICO is reset to zero here to allow for the case when the Mach line terminates on a grid point in the wake (as at 3), since a call to WAKE causes ICO = 1.

The computational procedure remains the same through the tenth Mach line (along A). MACHLN is called whenever IHAVEP = 0, and TOP/BOTTOM are called when IHAVEP = LCOUNT. GENFPT is called otherwise. At the completion of computation along (A), the integer switch variables are reset to

IHAVEP = 0
LCOUNT = 10
JLINE = 10
ICO = 0

and the other integer variables are as before,

LIMW = 10
NSTPTS = 4
NUMPI = 1
MAXI = 10

The computational procedure down Mach line (B) is the same until the grid point at (4), where,

IHAVEP = (NUMPI)(NSTPTS) = 4

and

JLINE = MAXI = 10

Satisfaction of this logical test envokes a call to NEWBLD to compute the perturbation quantities at the leading edge of the second blade. NEWBLD also also increments NUM so that NUM = 1 on return . Next, COEF1 is called to store the pressure difference at the first grid point on the second blade. Next,the integer switches are reset to,

ICO = 1
LCOUNT = IHAVEP + NGRDFN = 10
JLINE = IHAVEP = 4
IHAVEP = IHAVEP + 1 = 5
OLDL = IHAVEP = 5
NUMPI = NUM + 1 = 2
MAXI = NGRDFN + (NUMPI)(NSTPTS) = 14
NUM = NUM - 1 = 0

Computation down (B) continues until the grid point at (5), with GENFPT being called for all intervening grid points. For the first grid point after (4), ICO = 1, and was reset

equal to zero in GENFPT. At (5), IHAVE = LCOUNT = 10, and TOP/BOTTOM are called. COEF1 is called on return from BOTTOM, and GENFPT is called for the remaining grid points until IHAVEP = LIMIT. Then the integer switch variables are reset to

IHAVEP = 0
NUM = NUMOLD = 1
ICO = 0
JLINE = JLINE + 1 = 5

Since more than one blade has now been encountered, NUMOLD > 0, and it is necessary to reset the following integer switches so that the grid point on blade two can be identified.

LCOUNT = OLDL = 5
LIMW = (NUMOLD) (NSTPTS) + NEWDST = 14
OLDL = OLDL + 1 = 6

The computation now starts over at the initial left-running Mach line (Mach line C), where MACHLN is called for IHAVEP = 0. Thereafter, until the grid point at (6) where IHAVEP = LCOUNT = 5, GENFPT is called. At (6), ICO is set equal to one and TOP/BOTTOM are called. Next, COEF1 is called to store the pressure difference just computed. Upon return from COEF1, since NUM > 0, the integer switch variables must be reset so the grid point on the upper surface of the next blade (or wake, in this case) can be identified. Thus,

NUM = NUM - 1 = 0
LCOUNT + NGRDFN = 11
LIMW = LIMW - NSTPTS = 10

GENFPT is then called to compute the perturbation quantities at grid points until the program reaches (7), where IHAVEP = LCOUNT = 11. This time, IHAVEP > LIMW , which means that WAKE should be called instead of TOP/BOTTOM. Again, ICO is set equal to one before calling WAKE. After return from WAKE, COEF2 is called. As this is the first time that it has been called for this blade (i.e., this is the first grid point in the wake aft of blade one), the integration of the pressure distribution for the lift and moment is done.

Since this is the first call to COEF2 aft of blade one, IWRITE = 1 on return to the main program and the pressure distribution for this blade can be output, if desired. Next, a check is made to determine whether or not this was the last grid point in the flow field, or if IHAVEP = LIMIT. If not, the calculation loop is reentered, with ICO = 1. GENFPT is called to compute the remaining grid points until IHAVEP = LIMIT, where the integer switches are reset to

IHAVEP = 0
NUM = NUMOLD = 1
ICO = 0
JLINE = JLINE + 1 = 6

and as NUMOLD > 0,

LCOUNT = OLDL = 6
LIMW = (NUMOLD)(NSTPTS) + NEWDST = 14
OLDL = OLDL + 1 = 7

The computational procedure down Mach line (D) is the same as for (C). The call to COEF2 at (8) will result in an immediate return to the main program with IWRITE = 0 as this will be the second time (more accurately, not the first time) that COEF2 has been called for blade one. The computation proceeds in the same manner, down successive right-running Mach line, until reaching point (9) on Mach line (E), where,

IHAVEP = (NUMPI)(NSTPTS) = 8

and

JLINE = MAXI = 14

which initiates a call to NEWBLD for the assumed blade. Termination of the flow field calculations is indicated when

NUM > NUMBLD - 1 = 1

and

IHAVEP = LIMIT

at point (10). As (10) is also the first wake grid point aft of blade 2, COEF2 is called to compute the lift and moment for the second blade before terminating the flow field calculations.

D. SUBROUTINE COEF

Subroutine COEF is a multiple entry subroutine used to store the pressure difference distributions for each blade as it is encountered, to extrapolate for the pressure difference at the trailing edge of each blade, and to use these distributions to compute the lift and moment due to pitch and plunge. If the pressure distribution for a particular blade is required, output is obtained by issuing a write statement after return from COEF2 in the main program.

COEF is the first section of this subroutine. Called after return from INITAL, it initializes all arrays used in the subroutine and stores the pressure difference for the first grid point on the first blade. The pressure difference is then used to compute the incremental moment about the elastic axis.

COEF1 is called after every call to TOP/BOTTOM or NEWBLD, and stores the pressure difference with the real component referenced to the blade initial position, and the imaginary component referenced to the mean position, blade pitching leading edge up. Refer to Eqs. (III-70) - (III-75). The pressure differences are then used to compute the incremental moment resulting from each discrete pressure difference.

COEF2 is called after every call to WAKE. The first time this section is called, it initiates the calculation of the lift force and moment for the n^{th} blade, after extrapolating for the pressure difference at the trailing edge of the blade. At the same time, the integer switch

ID(n) is set equal to one. Subsequent calls to COEF2 for the n^{th} blade wake result in an immediate return to the main program, after setting IWRITE = 0. IWRITE is an integer switch variable used to indicate when the pressure distribution for the entire blade is complete. For a given blade, IWRITE = 1 only after the first call to COEF2, where the pressure difference at the trailing edge of the blade is determined. At all other times, IWRITE = 0. After the lift force and moment have been computed, the section of the matrix used for storage of the n^{th} blade pressure distribution can be reused for another blade. Upon return to MAIN, the pressure distribution for the n^{th} blade can be output, if desired, if IWRITE = 1.

E. OTHER SUBROUTINES

The remaining subroutines used to calculate perturbation quantities in the flow field are TOP, BOTTOM, GENFPT, NEWBLD, WAKE, and MACHLN, all called by the main program. Each subroutine solves the applicable finite difference equations derived in section III-B. Subroutine WAKE uses IBM subroutine SIMQ, a simultaneous equation solving routine, to compute the perturbation quantities at F_{22}. The solution procedure in each subroutine is the four step process described earlier in this section, and it is carried out first for pitch, and then for plunge in each subroutine. Each subroutine defines several constants, a few of which are common to both solutions.

Subroutine INPUT reads the input data, computes the compatible stagger angle, and the geometric cascade parameters used by other subroutines. Here the input data is checked to insure that available core size is not exceeded.

Subroutine INITAL uses the equations derived for the initial Mach lines to compute the perturbation quantities at the leading edge of the first blade. In addition, JLINE, LCOUNT and IHAVEP are initialized.

Finally, subroutine COMPXY computes the grid point coordinates in the characteristic mesh in terms of the the geometry computed in INPUT and shown in figure 13. The calculation procedure for points on the right-running Mach line is

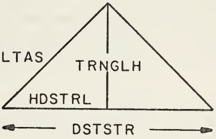


Figure 13. Characteristic mesh geometry.

X(I) = X(I) + HDSTRL

Y(I) = Y(I) - TRNGLH

The first grid point on a right-running Mach line is

X(1) = X(1) + HDSTRL

Y(1) = Y(1) + TRNGLH

Actually, the only coordinate used by the program is X(I).

Subroutine FLUTER is called for two degree of freedom flutter solutions when the input variable IFLUT = 1, and uses the equations derived in Section IV, where the single degree of freedom equations were given also. Flutter investigation is quite expensive in terms of computer time, especially if the program is designed to iterate to an actual flutter speed, subject to some error tolerance. To minimize the 'first blade effect', the flutter calculations must be conducted for a relatively high numbered blade where

the magnitude of the lift forces and moments have more nearly approached an assymptotic value. This must be done because an actual turbomachine does not have a 'first' blade and the aerodynamic forces on the blades are assumed to be periodic. To determine a critical flutter speed for a given cascade geometry and Mach number, the minimum flutter speed must be determined, which requires an iteration on the interblade phase angle as well. Since each iteration through the flutter routine requires a recomputation of the entire flow field, it is apparent that computational time mounts rapidly.

In an attempt to reduce the amount of computer time required for the flutter investigation, the computer program is set up to compute and output the roots of the flutter determinant versus the reciprocal of the reduced frequency $(1/k)$. These results are then plotted by hand, using the Theodorsen method to find a flutter solution.

$(1/k)$ is incremented by an amount STEP, an input variable. If, for a given value of $(1/k)$, the real or imaginary component of the solution is negative the program will terminate, as the dimensionless flutter frequency would be imaginary. However, if the discriminant of the quadratic for $\text{Re}\{X\}$ is negative, then $(1/k)$ is first decreased by STEP, and then increased by one-fifth that amount. The flow field is then recomputed, and FLUTER called again. Note that this procedure assumes that the reduced frequency used for the previous solution produced a positive real root. As the real component of the flutter determinant is quadratic, a negative discriminant indicates that the value of $(1/k)$ is past the vertex of the parabolic curve of $\sqrt{\text{Re}\{X\}}$ in the solution plane (see figure 9, Section IV). Each time through the flutter subroutine, the square root of the three roots of the flutter determinant and the corresponding value of $(1/k)$ are output. Because of the manner in which the

case of a negative real root is handled, and also because the minimum flutter speed occurs at the smallest value of (1/k), the program should initially be given a reduced frequency on the order of three or greater. In all cases investigated in this report, all three of the flutter determinant roots were positive for reduced frequencies of that magnitude.

F. INPUT AND OUTPUT

Subroutine INPUT reads all input data from two Namelist input records, NAM1 and FLTR. The input variables for NAM1 are,

NGRDFN - the grid fineness ratio (ϵ). The number of increments along a Mach line between two blades.

FSTRMN - the freestream Mach number.

REDFRQ - the reduced frequency, defined in Eq. (II-39).

XSUBO - the elastic axis position (x_0)

TNWDST - vertical distance between adjacent blades.

STGANG - the input stagger angle (β). The compatible stagger is computed in INPUT.

FAZE - the interblade phase angle (in degrees).

NUMBLD - the number of blades (N_{bld}) in the cascade.

IPRES1 - number of the blade user desires specific information about, i.e., pressure distribution, lift force and moment values, etc.

IPRES2 - same as IPRES1.

Variables in Namelist FLTR are all used in the flutter routine. They are,

RMUU - the blade density parameter (μ).

XSUBA - location of the center of gravity

measured from the elastic axis, positive aft (x_α).

RSUBA - the blade radius of gyration (r_α).

WHWA - ratio of natural frequency in plunge to the natural frequency in pitch ($\frac{\omega_h}{\omega_\alpha}$)

STEP - increment for (1/k).

IFLUT - set equal to one (IFLUT=1) if a flutter solution is desired.

The input data for the sample program discussed in part C of this section would be:

```
&NAM1
NGRDFN = 6,
NUMBLD = 2,
FAZE = 50.,
XSUBO = 0.5,
TNWDST = .33824,
STGANG = 62.8,
IPRES1 = 1,
IPRES2 = 2,
&END
&FLTR
RMUU = 500.,
XSUBA = 0.,
RSUBA = 0.5,
WHWA = 0.707,
STEP = 0.05,
IFLUT = 0,
&END
```

Each card begins in column 2, and the comma follows immediately after the last digit. Spaces between the last digit and the comma are interpreted by the computer as zeros.

Output can be modified at the users discretion. With the program as listed after Appendix B, pressure distributions, lift forces and moment and the magnitude and phase of the complex lift and moment for each blade can be output. In addition, if appropriate 'write' statements are put in each of the subroutines, the entire flow field may be printed. This option produces sizeable quantities of output, and is useful primarily for program debugging. The output from

flutter solution runs was described in Section V-D.

G. GENERAL

Because the program computes a compatible stagger angle, the cascade geometry actually used in the computer program may be slightly different than that input by the user. For large values of the grid fineness ratio, ϵ , the differences are insignificant. However, it is normally preferable to use a coarser grid size to minimize computer time. As a result, if a specific cascade is to be input, then an iteration on ϵ should be done to find the smallest value of ϵ which produces the compatible stagger angle that is in closest agreement with the input stagger angle. If this procedure is not followed, the cascade computed for will be slightly different than desired, and therefore, the reflection points on the blades will be different than expected.

The computer program requires approximately 175K bytes of storage (G compiler) or 135K bytes of storage (H compiler) when run on an IBM 360/67 computer. Compile time is approximately twenty-five seconds. Actual CPU time per run is dependent on the input geometry, grid fineness and number of blades. Sample times are given for each of the results discussed in Section VI.

Initial program checkout and verification can be accomplished in several ways. By setting the reduced frequency equal to zero, the well known result for the lift on a flat plate at unit angle of attack is obtained (for the first blade). Also, tabulated results appearing in the report by Garrick and Rubinow [Ref. 9] for single blades require less than five seconds of computer time to generate using a grid fineness of 100 or less.

VI. RESULTS

Initial work was done for single degree of freedom torsion. All cases discussed in this section were run with zero structural damping (g = 0). Pressure distributions and flutter solutions were computed and compared with Snyder [Ref. 19], who used a finite difference procedure developed by Verdon [Ref. 23]. Brix [Ref. 2] showed that considerable discrepancy existed between the pressure distributions given by Snyder and those computed by the method of characteristics, especially in the region of reflection points. As stated by Verdon [Ref. 25, p24],

> The finite difference marching procedure used . . . does not recognize pressure discontinuities as such, but only approximates them by sharp changes in the pressure distribution. This numerical approximation also gives rise to the ripples which appear in the predicted pressure distributions downstream of the discontinuities.

The flutter comparisons shown in figures 13 and 14 are for cascades C, D and E, described below.

	Cascade C		Cascade D		Cascade E	
	β = 67		β = 60		β = 70	
M	ϵ	Time	ϵ	Time	ϵ	Time
1.3	24	70	35	75	26	85
1.4	27	50	39	60	20	45
1.5	27	45	40	50	22	45
1.6	27	35	31	25	20	25
1.7	28	30	54	40	22	30
1.8	28	25	38	20	24	30

The entry in the Time column is the actual CPU time, in seconds, required for one iteration through the flow field for a ten blade cascade. Figures begin on page 91.

All three cascades have a solidity of 1.33 and a

mid-chord pitch axis (x_0). The grid fineness, ϵ, was varied in order to be able to match the geometry of each cascade as closely as possible (refer to V-G), while maintaining at least twenty-five discrete grid points along each blade. Both results assumed zero structural damping (g_α). In addition, the method of characteristics solution used an assumed value of 500 for the blade density parameter μ, and a radius of gyration, r_α, of 0.5. The value for μ was felt to be reasonable for modern turbomachinery blading [Ref. 5]. These figures, 14 and 15 show the reduced frequency which occurs at the minimum flutter speed, k_{crit}, and the interblade phase angle at that point, δ_{crit}, versus Mach number. Method of characteristics results are based on the tenth blade. Agreement is not exact. However, in light of the previous comment on the computational scheme used by Snyder, the agreement is felt to be reasonable.

In figures 16 to 20, similar results are shown for Cascade C with elastic axis as a parameter. Again, $\mu = 500$, and $r_\alpha = 0.5$.

Figures 21 to 24 present flutter stability boundaries for Cascade C. These are plots of elastic axis position versus Mach number for which $M_4 = 0$. The shaded portion of each graph is unstable, i.e., $M_4 < 0$. The analysis was made for 30° increments in the interblade phase angle, δ. Figures 21 and 22 showed instabilities at only the values of shown. In figure 23, the range of interblade phase angles over which instabilities existed increased, and in figure 24, instabilities occurred for all values of examined.

Thus, these four figures show that as the reduced frequency is increased (1/k decreased), cascade operation becomes more stable.

When conducting flutter investigations, it is important to find the highest value of reduced frequency for which the cascade becomes unstable, as this will produce the lowest flutter speed for a given geometry and interblade phase angle. Figure 25 shows the variation of M_4 with (1/k). As can be seen, if the step size in (1/k) is too large, the first point at which $M_4 < 0$ could be missed (note inset in the figure). As a result, the reduced frequency at which $M_4 = 0$ at B would result in an incorrect (too high) flutter speed. Also, figure 25 shows the condition where M_4 just crosses the axis. When the phase angle was slightly decreased, the curve of M_4 was not quite tangent to the axis at A, but crossed the axis near B. Figure 26 shows the resulting jump in the flutter speed and reduced frequency at flutter.

A limited amount of investigation was conducted to determine the dependence of flutter speed on blade number. Although M_4 is quite oscillatory for the first few blades and does not approach an asymptotic value until after the tenth or fifteenth blade, preliminary results, based on a limited number of cases, indicated that the flutter speed and frequency were fairly constant. An example is shown in figure 27 for Cascade D with $M = 1.7$, $\delta = -45$, and $\epsilon = 25$.

Comparison with results by Verdon [Ref. 25] for an infinite cascade was made using the two degree of freedom

computer program listed at the end of this report. The cascades A and B referred to in the figures are shown and described in figure 28. Pitch results are always for the fifteenth blade, and plunge results, for the fourteenth. In figures 29 through 49, the time dependence is in terms of the compressible reduced frequency as defined by Verdon, where $k_c = k \frac{M}{M^2-1}$. Corresponding to Verdon's notation, $\mathrm{Im}\{M\} > 0$ implies unstable operation for the pitch case. Plunge oscillation was always found to be stable. In addition, figure 29 shows convergence of the norm of M (norm $= |M_\alpha| = \sqrt{M_\alpha \bullet M_\alpha}$) versus blade number for four different values of δ. These phase angles are all in the region for which Verdon was unable to obtain a solution. The curves are shown as being continuous for clarity of presentation, but results were only obtained at discrete integer values of blade number, of course. Each curve required approximately nineteen minutes of computer time.

Figures 30 through 33 are plots of the real and imaginary component of the pressure distribution acting on the upper and lower surfaces of the blades for the two cascades. Figures 34 through 37 show the pressure difference across the blade (lower - upper) for two different interblade phase angles. Agreement was excellent.

Figure 38 defines the symbols used in figures 39 through 49, where lift and moment loops are obtained by plotting the real and imaginary components of the lift force and the moment with the interblade phase angle as a parameter. The method of characteristics was able to calculate solutions throughout the range of interblade phase angles, $0 < \delta <$

360. Again, excellent agreement was obtained.

In figures 30 through 49, results for cascade A required 48-50 seconds of CPU time per fifteenth blade solution, and for cascade B, 100-110 seconds per fourteenth blade solution.

The last figure, 50, shows a flutter solution for cascade A assuming the following parameters: μ = 500, r_α = 0.5, x_α = 0, and $\frac{\omega_h}{\omega_\alpha}$ = .707.

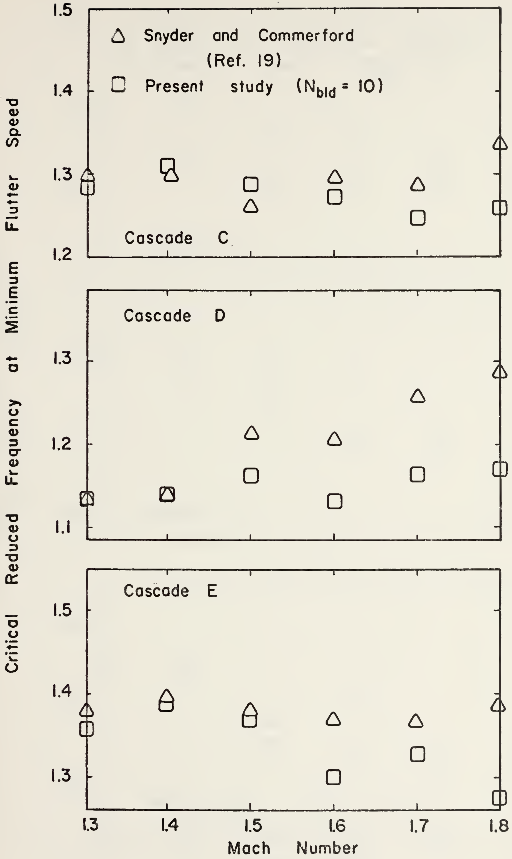


Figure 14. Reduced frequency at minimum flutter speed versus Mach number (μ=500, r_α=0.5).

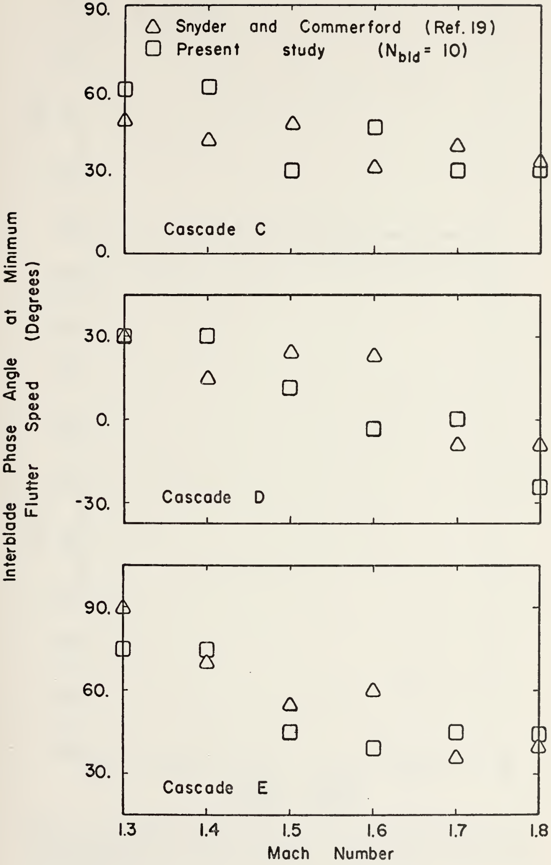


Figure 15. Interblade phase angle at minimum flutter speed versus Mach number (μ=500,r_α=0.5),

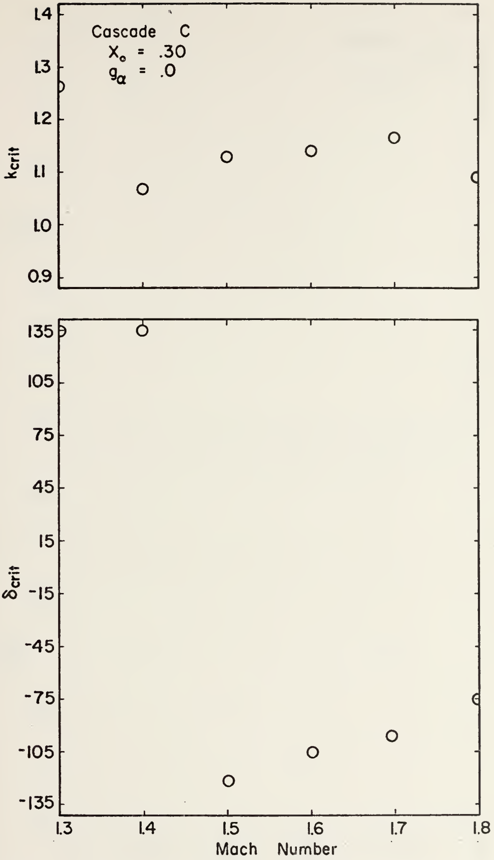


Figure 16. Reduced frequency (k_{crit}) and interblade phase angle (δ_{crit}) at minimum flutter speed versus Mach number (μ=500,r_α=0.5).

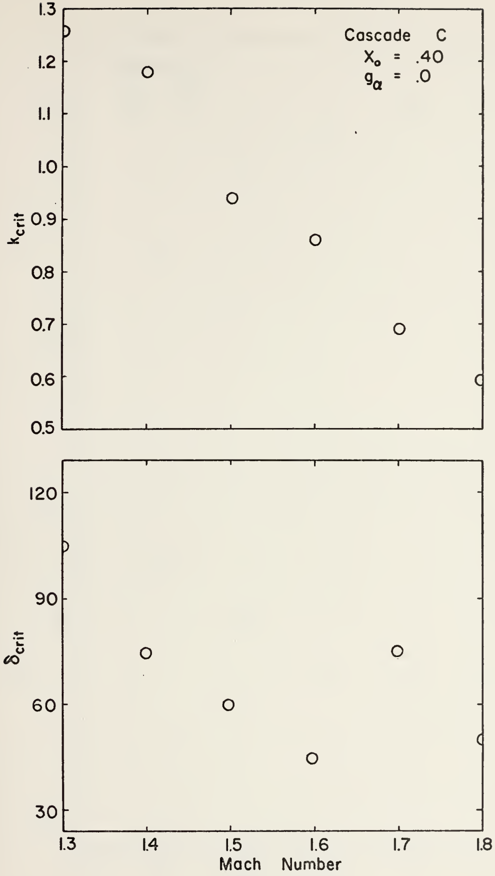


Figure 17. Reduced frequency (k_{crit}) and interblade phase angle (δ_{crit}) at minimum flutter speed versus Mach number (μ=500, r_α=0.5).

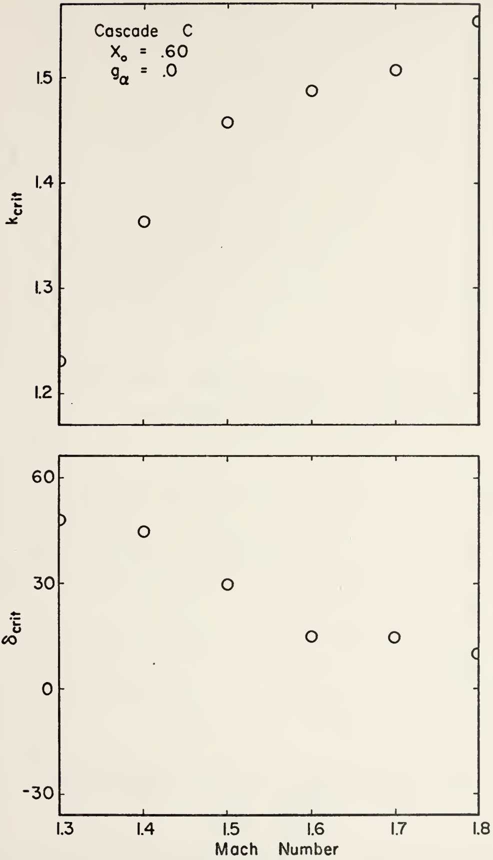


Figure 18. Reduced frequency (k_{crit}) and interblade phase angle (δ_{crit}) at minimum flutter speed versus Mach number (μ=500, r_α=0.5).

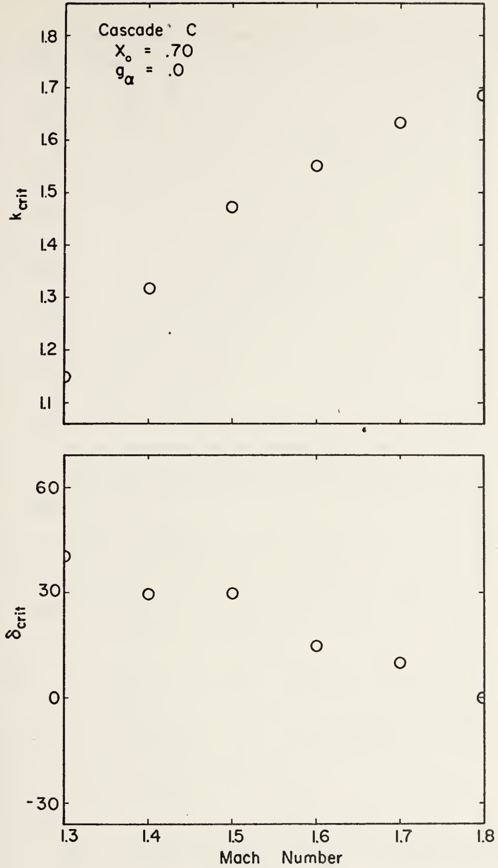


Figure 19. Reduced frequency (k_{crit}) and interblade phase angle (δ_{crit}) at minimum flutter speed versus Mach number (μ=500, r_α=0.5).

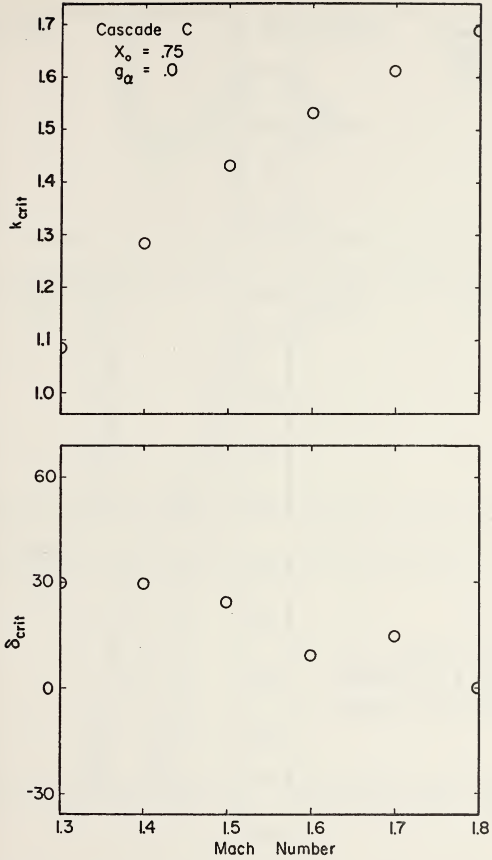


Figure 20. Reduced frequency (k_{crit}) and interblade phase angle (δ_{crit}) at minimum flutter speed versus Mach number (μ=500,r_α=0.5).

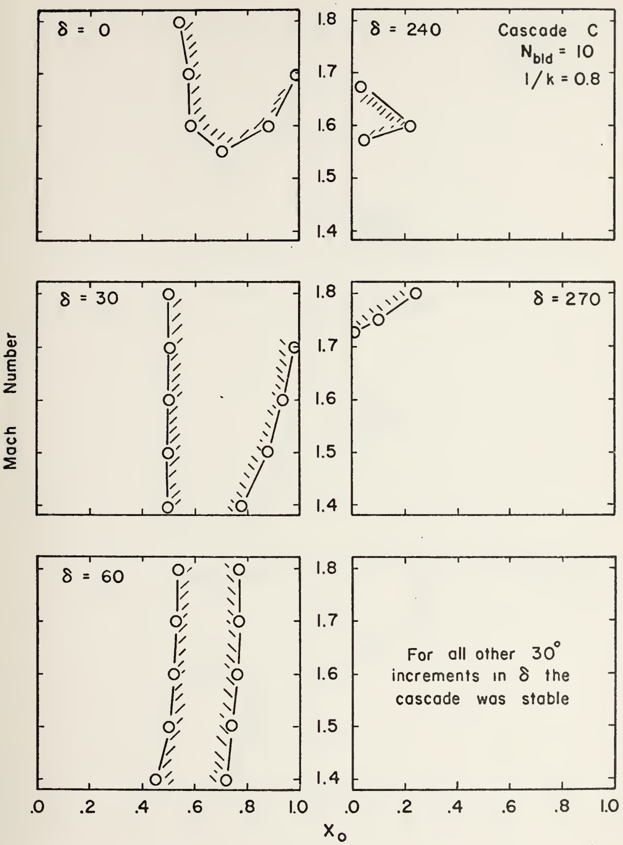


Figure 21. Flutter boundaries versus elastic axis position (shaded portion is unstable).

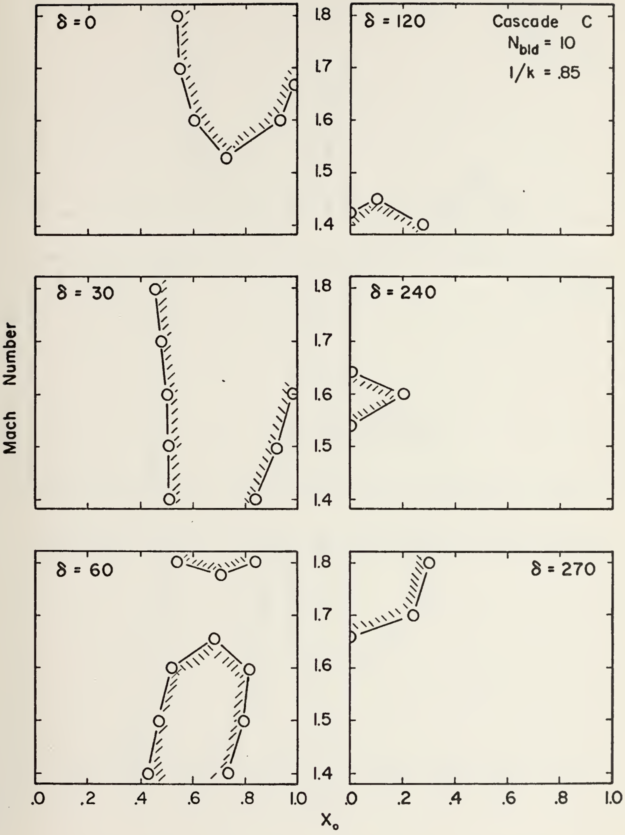


Figure 22. Flutter boundaries versus elastic axis position (shaded portion is unstable).

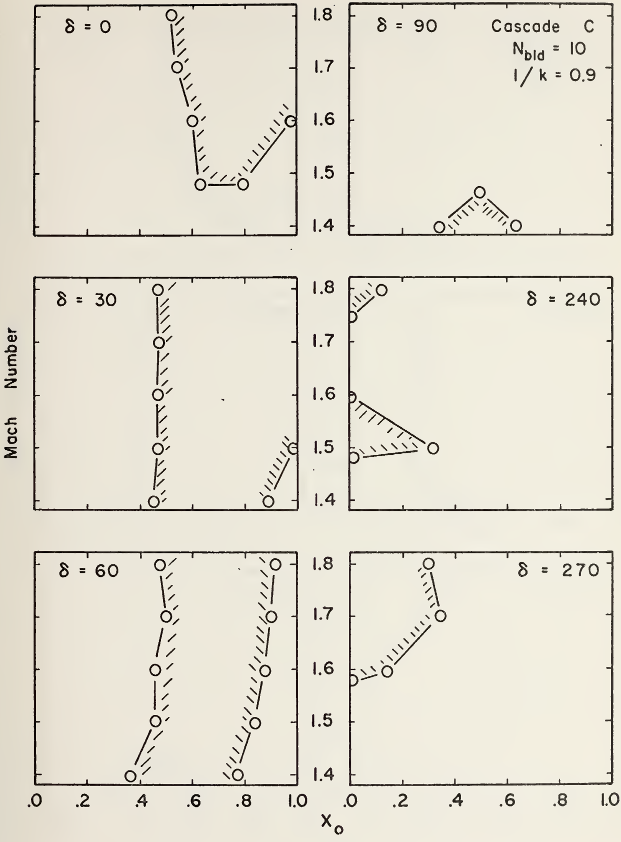


Figure 23. Flutter boundaries versus elastic axis position (shaded portion is unstable).

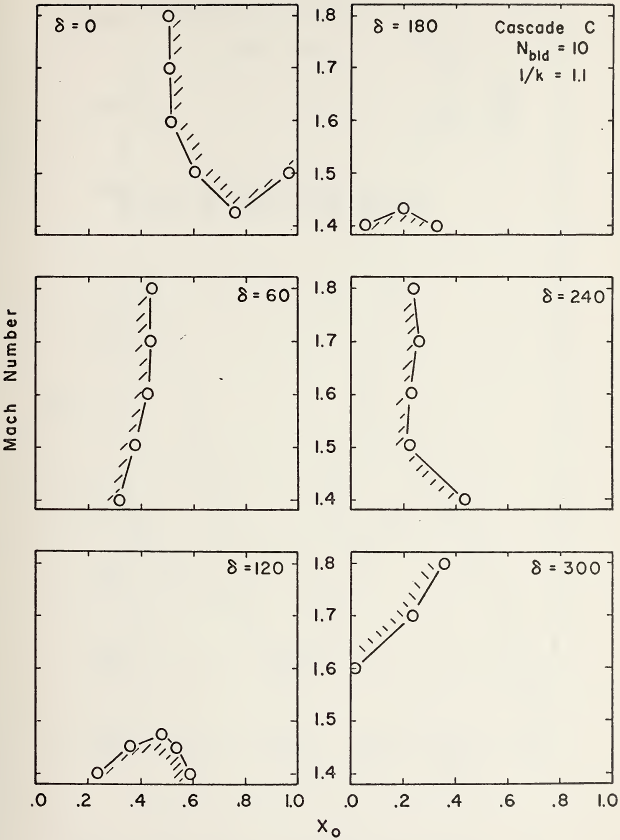


Figure 24. Flutter boundaries versus elastic axis position (shaded portion is unstable).

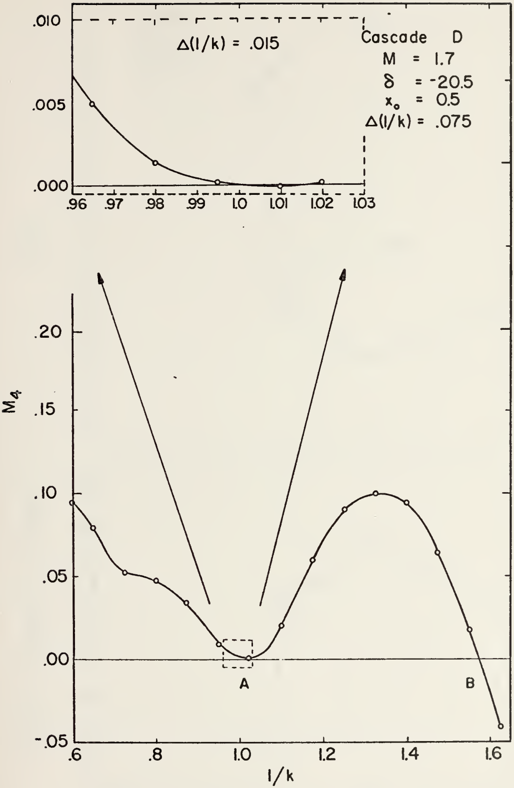


Figure 25. Oscillation of the imaginary component of the moment due to pitch (M_4) with reduced frequency.

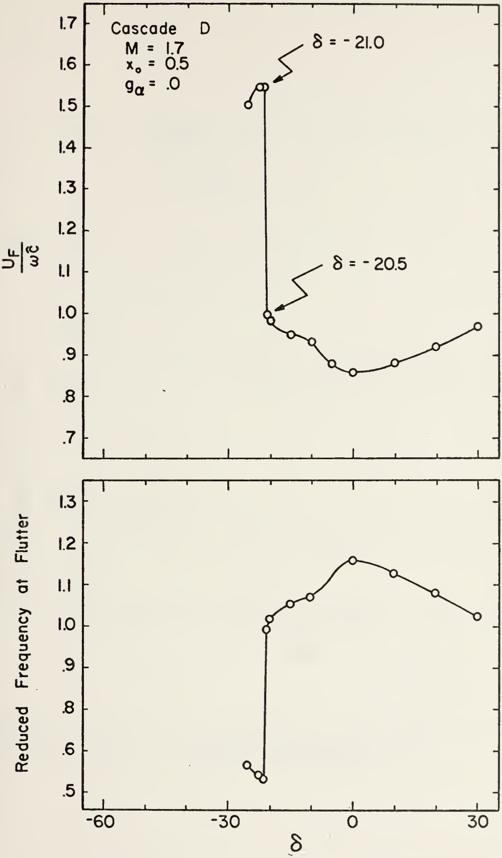


Figure 26. Variation of flutter speed and reduced frequency at flutter with interblade phase angle (μ=500,r_α=0.5).

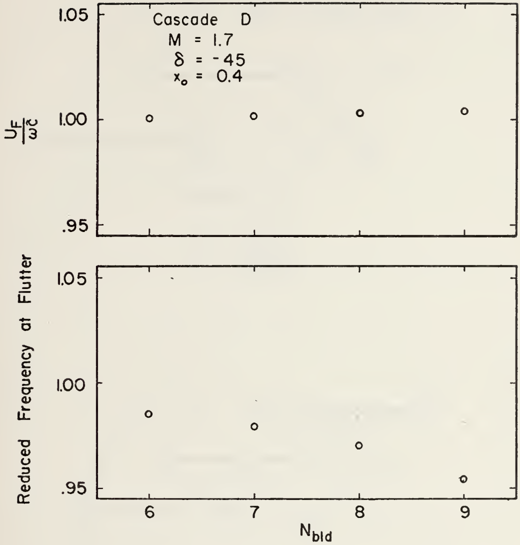
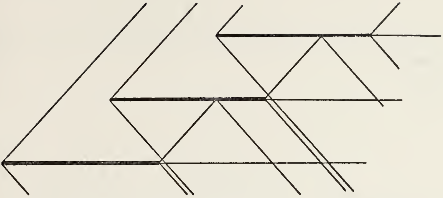
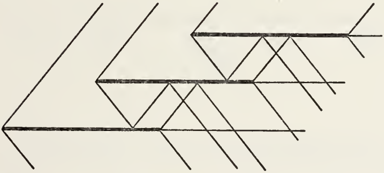


Figure 27. Non-dimensional flutter speed and reduced frequency at flutter versus blade index (μ=500,r_α=0.5).



Cascade A

$M = 1.345$
$\beta = 59.5$
$\sigma = 1.268$



Cascade B

$M = 1.281$
$\beta = 63.4$
$\sigma = 1.490$

Figure 28. Cascade A and B geometry.

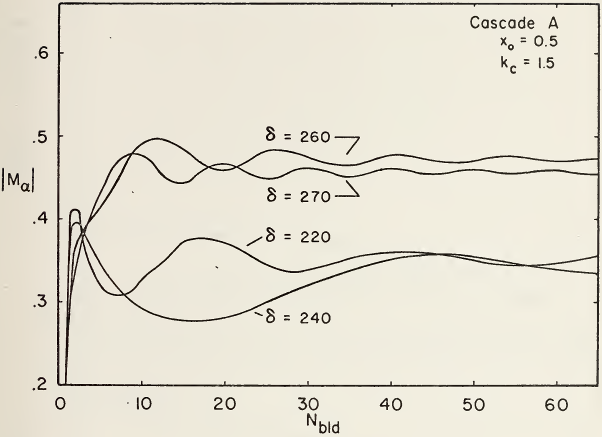


Figure 29. Convergence of the norm of the moment due to pitch oscillation with blade index as a function of the interblade phase angle.

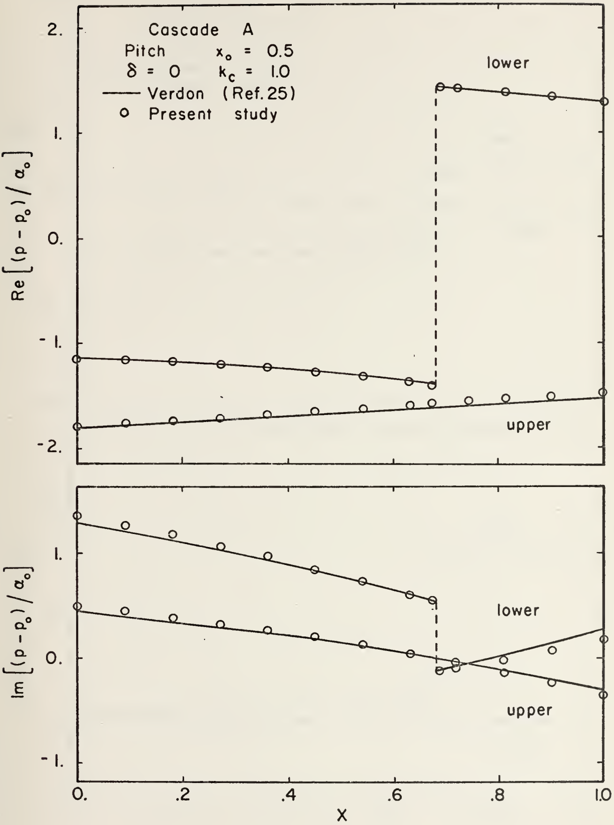


Figure 30. Comparison of the pressure distribution resulting from pitch oscillation of the fifteenth blade with the infinite cascade results of Verdon (Cascade A).

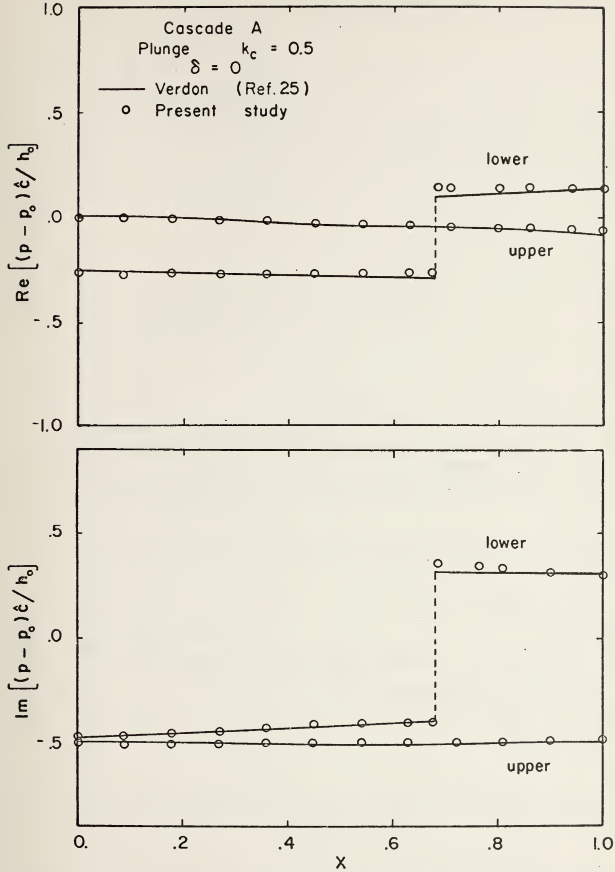


Figure 31. Comparison of the pressure distribution resulting from plunge oscillation of the fourteenth blade with the infinite cascade results of Verdon (Cascade A).

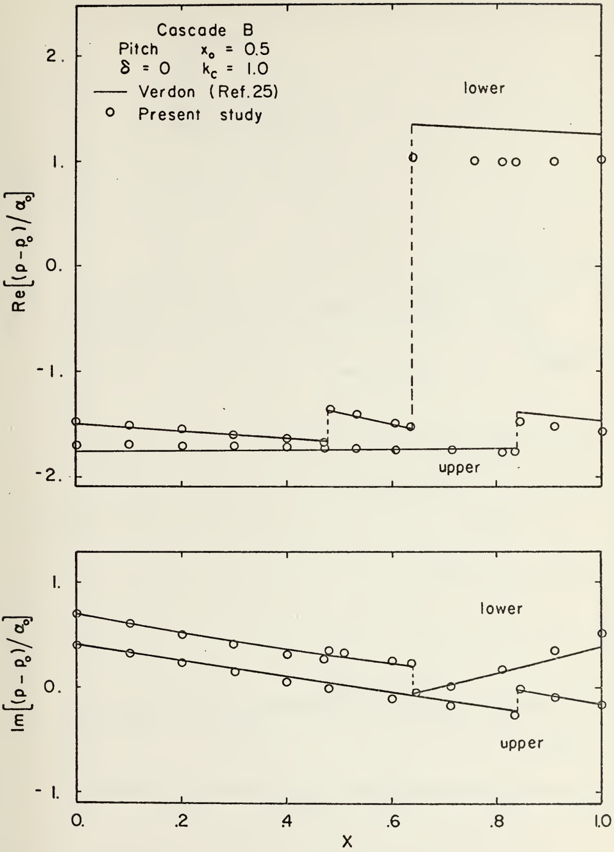


Figure 32. Comparison of the pressure distribution resulting from pitch oscillation of the fifteenth blade with the infinite cascade results of Verdon (Cascade B).

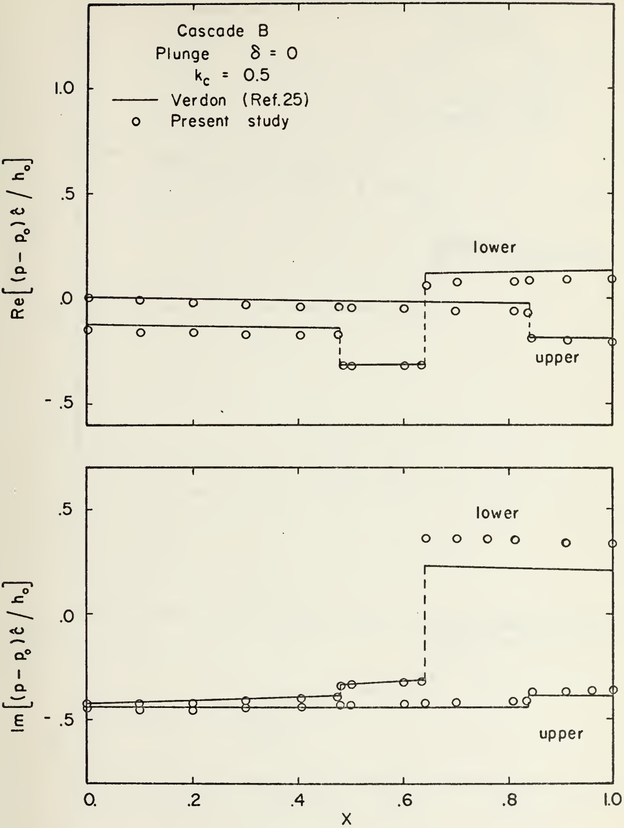


Figure 33. Comparison of the pressure distribution resulting from plunge oscillation of the fourteenth blade with the infinite cascade results of Verdon (Cascade B).

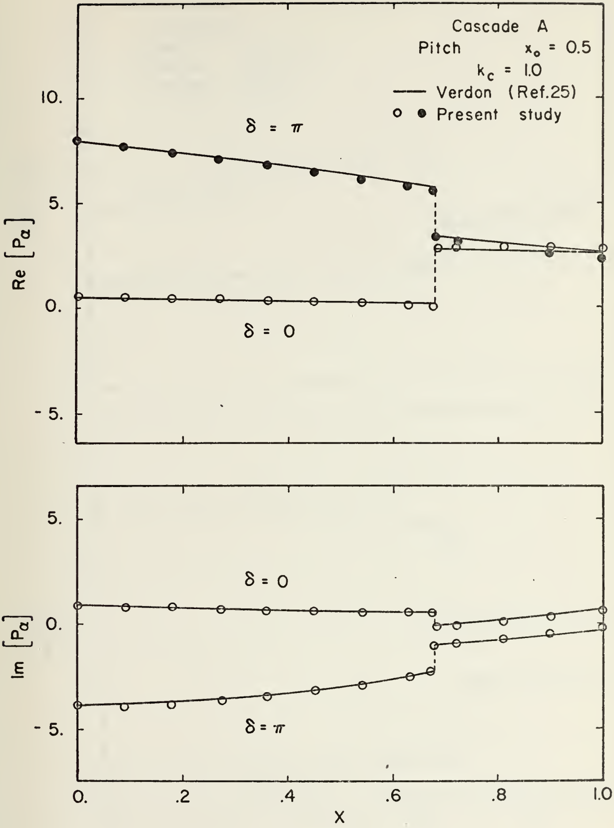


Figure 34. Comparison of the pressure difference distribution (lower - upper) resulting from pitch oscillation for the fifteenth blade with the infinite cascade results of Verdon (Cascade A).

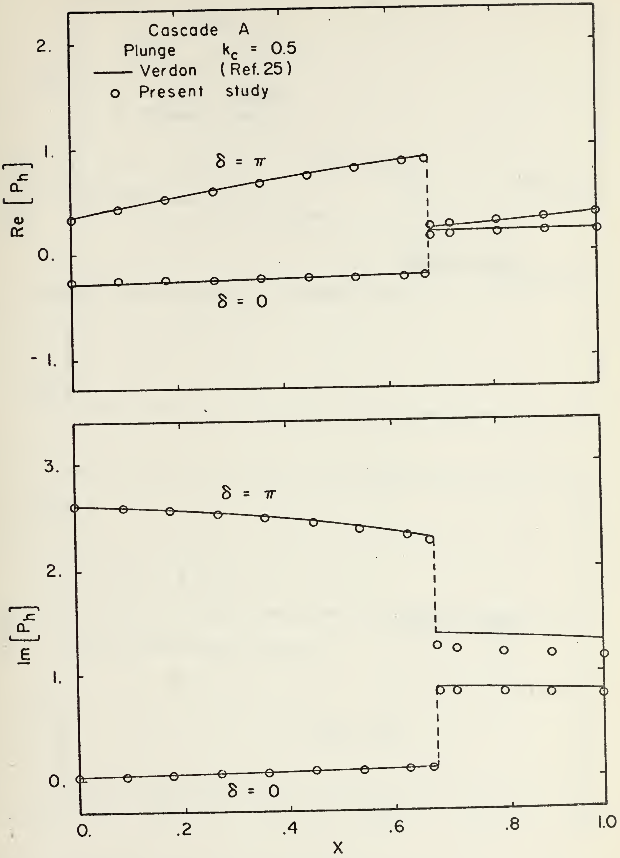


Figure 35. Comparison of the pressure difference distribution (lower - upper) resulting from plunge oscillation of the fourteenth blade with the infinite cascade results of Verdon (Cascade A).

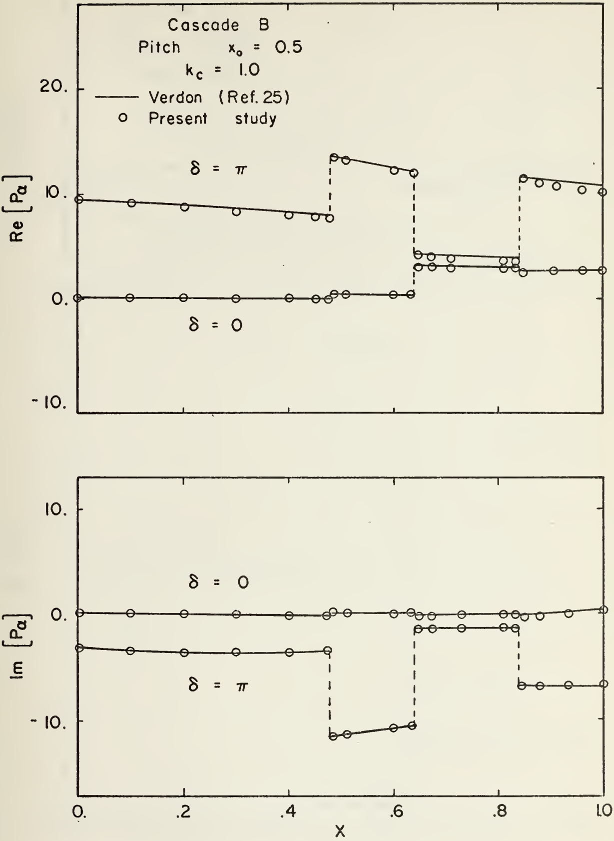


Figure 36. Comparison of the pressure difference distribution (lower - upper) resulting from pitch oscillation for the fifteenth blade with the infinite cascade results of Verdon (Cascade B).

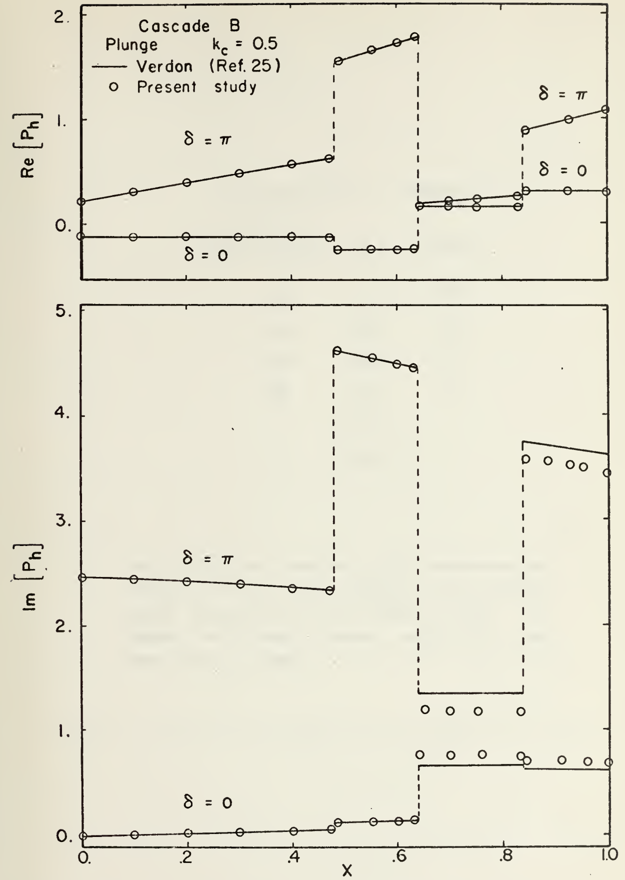


Figure 37. Comparison of the pressure difference distribution (lower - upper) resulting from plunge oscillation of the fourteenth blade with the infinite cascade results of Verdon (Cascade B).

EXPLANATION of SYMBOLS

Method of Characteristics	δ	Verdon
■	0	□
▲	90	△
▼	180	▽
⬢	270	⬡
●	$\delta + 30$	
○	$\delta + 10$	

The solid curve connects points determined by Verdon (Ref. 25). The dashed curve connects points determined in the present study. For regions of δ with common results, the solid curve only is shown.

Figure 38. Definition of symbols used in figures 39 through 49.

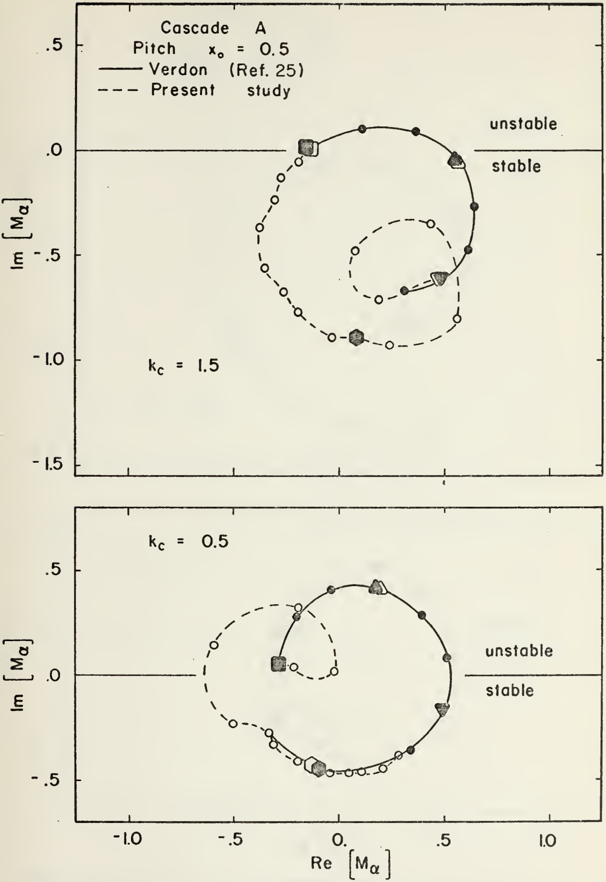


Figure 39. Comparison of the moment coefficients due to pitch oscillation for the fifteenth blade with the infinite cascade results of Verdon (Cascade A).

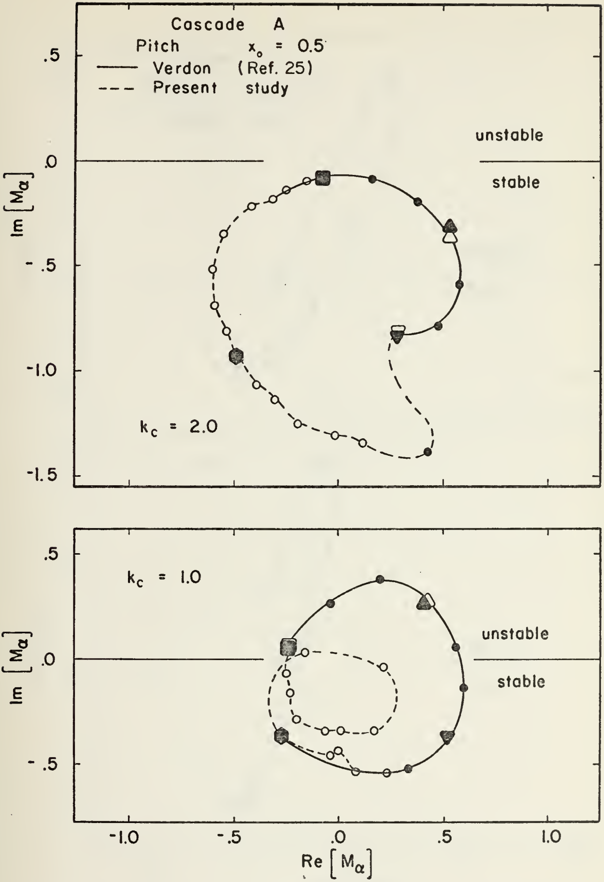


Figure 40. Comparison of the moment coefficients due to pitch oscillation for the fifteenth blade with the infinite cascade results of Verdon (Cascade A).

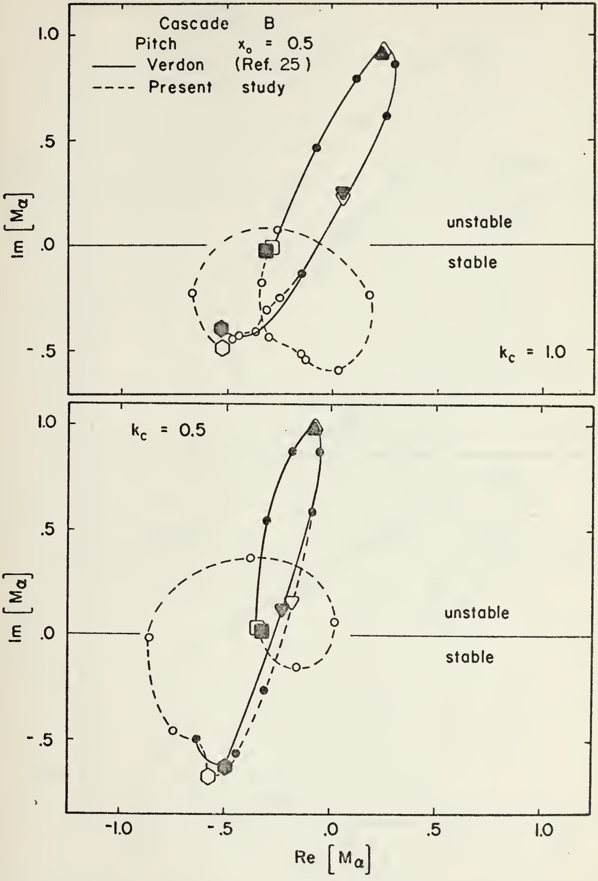


Figure 41. Comparison of the moment coefficients due to pitch oscillation for the fifteenth blade with the infinite cascade results of Verdon (Cascade B).

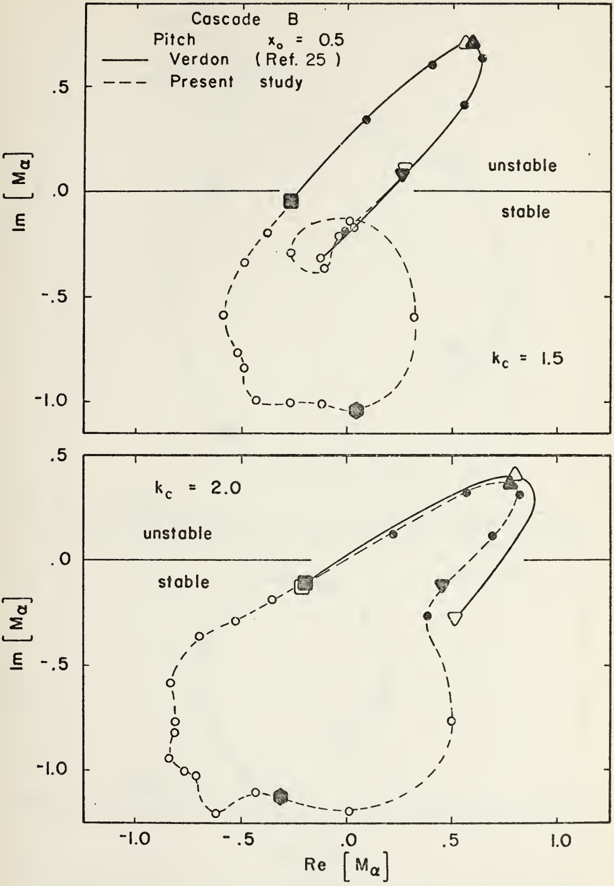


Figure 42. Comparison of the moment coefficients due to pitch oscillation for the fifteenth blade with the infinite cascade results of Verdon (Cascade B).

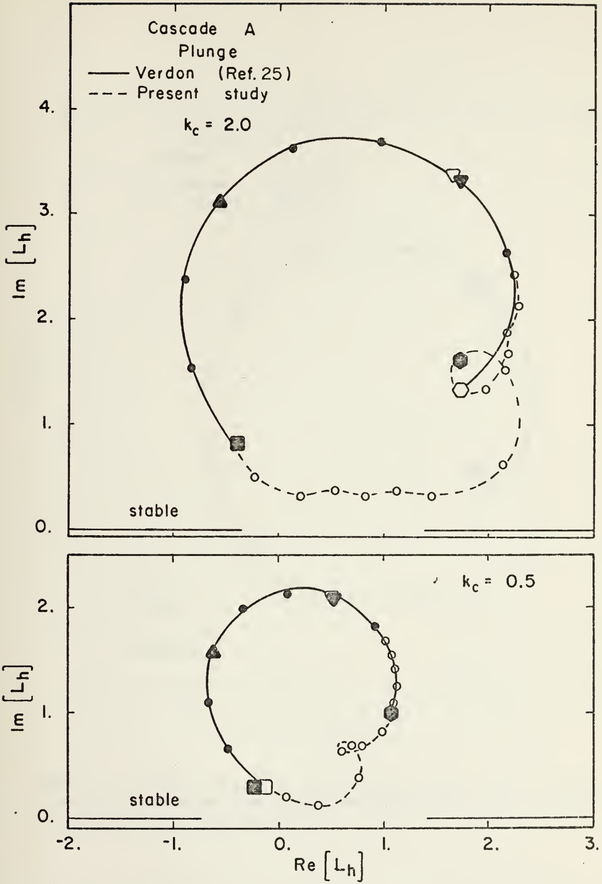


Figure 43. Comparison of the lift coefficients due to plunge oscillation for the fourteenth blade with the infinite cascade results of Verdon (Cascade A).

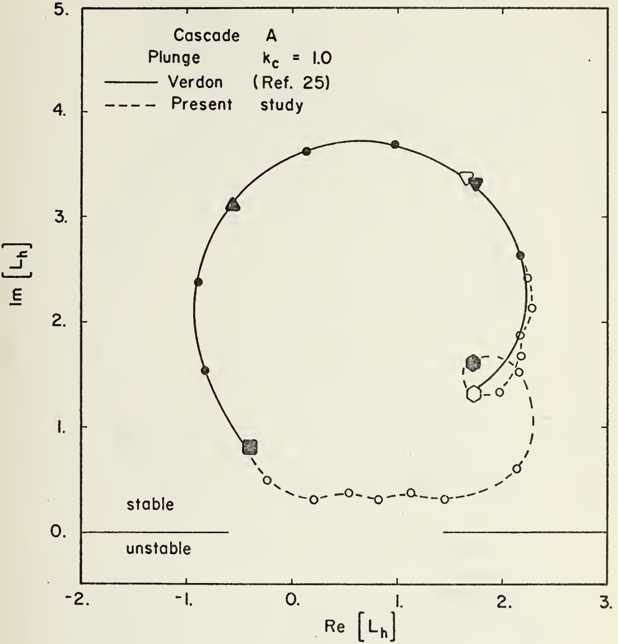


Figure 44. Comparison of the lift coefficients due to plunge oscillation for the fourteenth blade with the infinite cascade results of Verdon (Cascade . A).

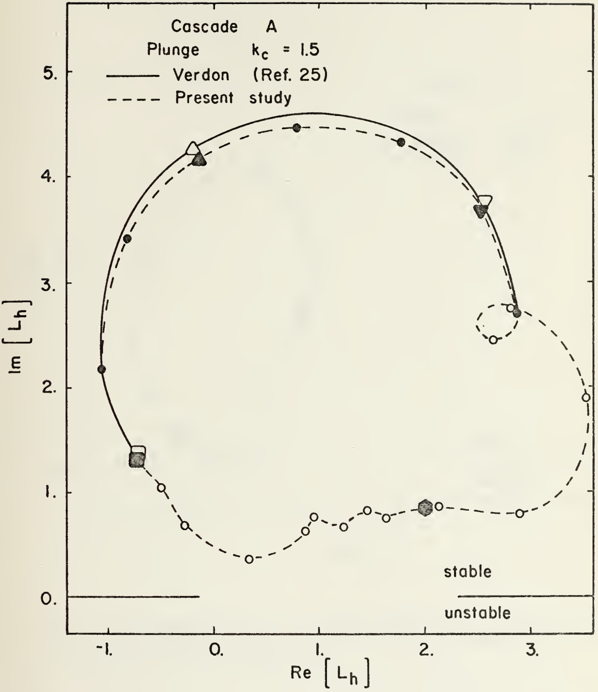


Figure 45. Comparison of the lift coefficients due to plunge oscillation for the fourteenth blade with the infinite cascade results of Verdon (Cascade A).

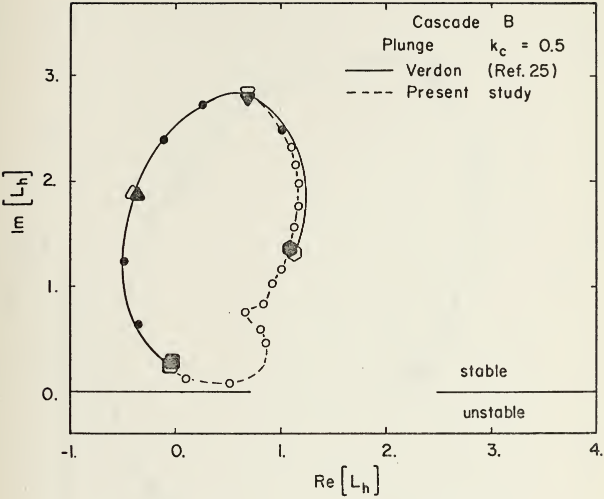


Figure 46. Comparison of the lift coefficients due to plunge oscillation for the fourteenth blade with the infinite cascade results of Verdon (Cascade B).

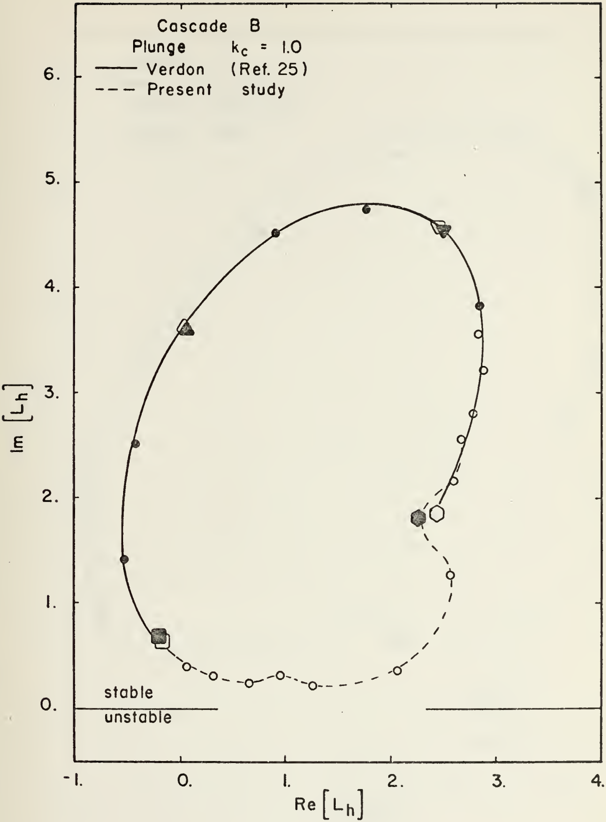


Figure 47. Comparison of the lift coefficients due to plunge oscillation for the fourteenth blade with the infinite cascade results of Verdon (Cascade B).

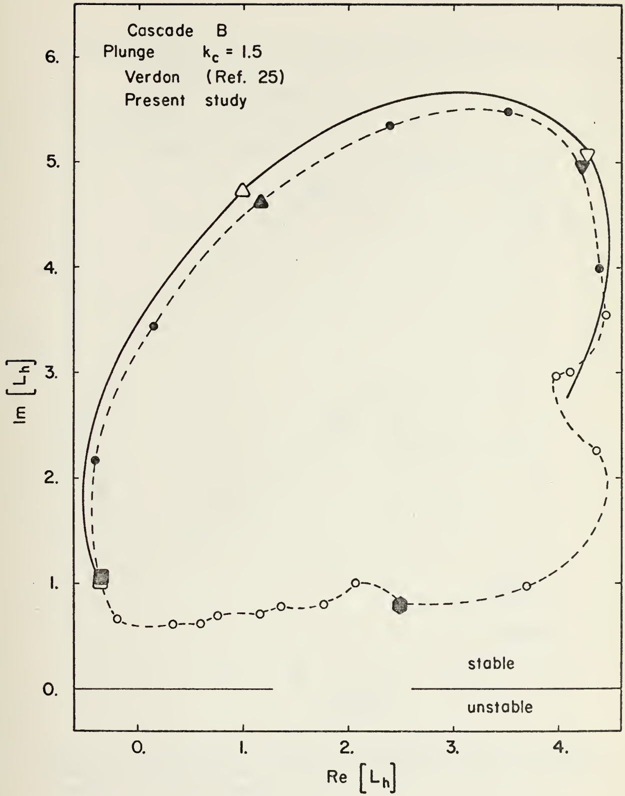


Figure 48. Comparison of the lift coefficients due to plunge oscillation for the fourteenth blade with the infinite cascade results of Verdon (Cascade B).

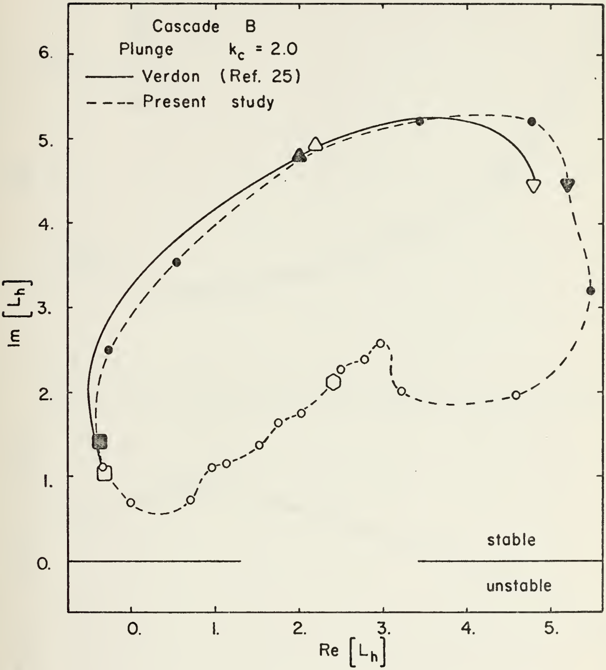


Figure 49. Comparison of the lift coefficients due to plunge oscillation for the fourteenth blade with the infinite cascade results of Verdon (Cascade B).

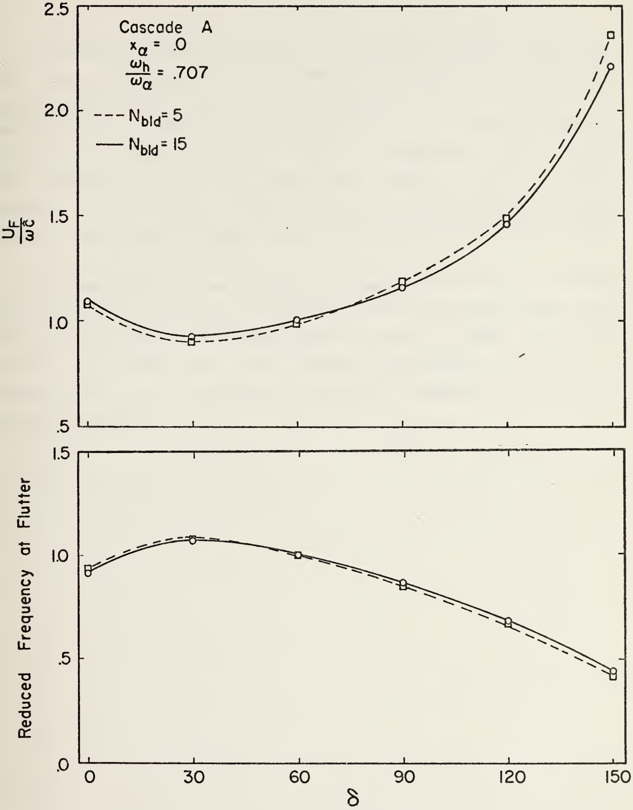


Figure 50. Non-dimensional flutter speed and reduced frequency at flutter versus interblade phase angle for Cascade A (μ=500,r_α= 0.5).

VII. CONCLUSIONS

A method of characteristics procedure has been developed for the solution of the complete linearized unsteady aerodynamics of a flat plate cascade having a subsonic leading edge locus in a supersonic flow field. Pressure distributions were computed and the resulting lift forces and moments used to investigate flutter boundaries and stability limits for a two-dimensional finite cascade, which can have arbitrary stagger angle and interblade phase angle.

Agreement with the infinite cascade results by Verdon were excellent. However, further verification and comparison with new methods being developed by others [Refs. 12,20,26] should be conducted. Of primary importance would be the development of a periodic solution applicable for the infinite cascade.

VII. CYLINDRICAL SHELL

This developement is closely based on a previous paper by Platzer, Brix and Webster [Ref. 17].

A. PROBLEM FORMULATION

Consider the supersonic flow of an inviscid, adiabatic gas through a circular cylinder whose axis is aligned with the freestream, and whose surface is rigid upstream of the origin. The cylinder geometry is shown in figure 51.

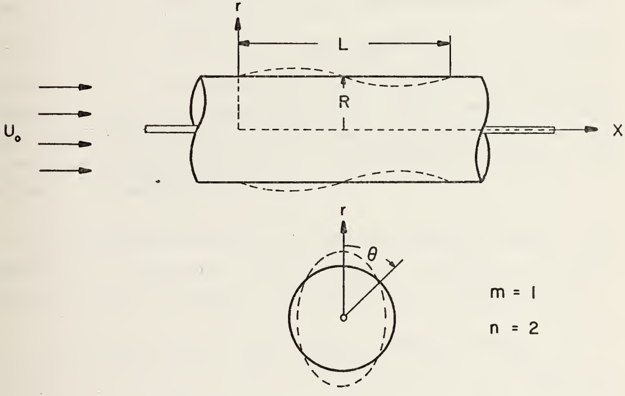


Figure 51. Cylinder Geometry

Downstream of the origin, a flexible panel of length L is assumed to undergo small amplitude vibrations. The governing equation for this problem is the linearized, unsteady velocity potential equation in cylindrical

coordinates, $\phi(X,R,\theta,T)$,

$$(1-M^2)\phi_{XX} + \phi_{RR} + \frac{1}{R}\phi_R + \frac{1}{R^2}\phi_{\theta\theta} - 2\frac{M}{c}\phi_{XT} - \frac{1}{c^2}\phi_{TT} = 0 \qquad \text{(VIII- 1)}$$

The linearized form of the pressure coefficient is,

$$\bar{C}_p = -\frac{2}{u_o^2}\phi_T - \frac{2}{u_o}\phi_X \qquad \text{(VIII- 2)}$$

The linearized boundary condition on the cylinder inner surface is

$$\phi_R \Big|_{r=R_o} = \frac{\partial \bar{h}}{\partial T} + u_o\frac{\partial \bar{h}}{\partial X}, \qquad 0 \leq X \leq L \qquad \text{(VIII- 3)}$$

The boundary condition at $r = 0$ is discussed separately. In Eq. (3), $\bar{h} = \bar{h}(X,\theta,T)$ is the equation for the panel surface. One other condition that must be satisfied is the Sommerfeld radiation condition; i.e., waves must propagate away from the source of disturbance. For a derivation of the above three equations, refer to Bisplinghoff, Ashley, and Halfman, "Aeroelasticity" [Ref. 1].

To non-dimensionalize the above equations, lengths are referred to flexible cylinder length and time to flexible cylinder length divided by the freestream velocity. Then,

$$(1-M^2)\phi_{xx} + \phi_{rr} + \frac{1}{r}\phi_r + \frac{1}{r^2}\phi_{\theta\theta} - 2M^2\phi_{xt} - M^2\phi_{tt} = 0 \qquad \text{(VIII- 4)}$$

$$\bar{C}_p = -2[\phi_t + \phi_x] \qquad \text{(VIII- 5)}$$

subject to,

$$\Phi_r \Big|_{r=R} = \frac{\partial \bar{h}}{\partial t} + \frac{\partial \bar{h}}{\partial x}, \qquad 0 \le x \le 1 \qquad \text{(VIII- 6)}$$

The non-dimensional shell deflection is assumed to be of the form,

$$\bar{h}(x,\theta,t) = Z(x)\cos(n\theta)\,e^{ikt} \qquad \text{(VIII- 7)}$$

where,

$$k = \frac{\omega L}{u_o} \qquad \text{(VIII- 8)}$$

and n is the circumferential mode number. Then the perturbation potential must be of the form,

$$\Phi(x,r,\theta,t) = \phi(x,r)\cos(n\theta)\,e^{ikt} \qquad \text{(VIII- 9)}$$

and the resulting equations are,

$$(1-M^2)\phi_{xx} + \phi_{rr} + \frac{1}{r}\phi_r + \left(k^2M^2 - \frac{n^2}{r^2} \right)\phi - i2kM\phi_x = 0 \qquad \text{(VIII-10a)}$$

$$C_p = -\,2[\phi_x + ik\phi] \qquad \text{(VIII-10b)}$$

C_p is the pressure coefficient amplitude. The boundary conditions are,

$$\phi_r \Big|_{r=R_o} = Z_x + ikZ, \qquad 0 \le x \le 1 \qquad \text{(VIII-11a)}$$

The boundary condition at the axis is dependent on the circumferential mode number n. Thus,

$$\phi_r \Big|_{r=0} = 0, \quad \text{for } n \text{ even} \qquad \text{(VIII-11b)}$$

$$C_p \Big|_{r=0} = 0, \quad \text{for } n \text{ odd} \qquad \text{(VIII-11c)}$$

Eqs. (10) and (11) are the basic equations which will be used in the cylindrical shell analysis.

B. METHOD OF CHARACTERISTICS

For $M > 1$, Eq. (10) is hyperbolic and has real characteristics which satisfy the ordinary differential equation

$$(M^2-1)d^2r - d^2x = 0 \tag{VIII- 12}$$

For ds the differential arc length along a characteristic, $d^2s = M^2d^2r$, and,

$$\frac{dx}{ds} = \frac{\sqrt{M^2-1}}{M} \tag{VIII-13a}$$

$$\frac{dr}{ds} = \pm\frac{1}{M} \tag{VIII-13b}$$

The derivative along a characteristic of an arbitrary function $F(x,r)$ is,

$$\frac{dF}{ds} = \frac{\partial F}{\partial x}\frac{dx}{ds} + \frac{\partial F}{\partial r}\frac{dr}{ds} \tag{VIII- 14}$$

Then,

$$F_1 = \frac{dF}{ds_1} = \frac{\sqrt{M^2-1}}{M}F_x + \frac{1}{M}F_r \tag{VIII-15a}$$

$$F_2 = \frac{dF}{ds_2} = \frac{\sqrt{M^2-1}}{M}F_x - \frac{1}{M}F_r \tag{VIII-15b}$$

where ds_1 and ds_2 are the differential arc lengths along the left and right characteristics respectively. The second (cross) derivatives are,

$$-F_{12} = \frac{1}{M}[(1-M^2)F_{xx} + F_{rr}] = -F_{21} \qquad \text{(VIII- 16)}$$

From Eq. (15),

$$F_x = \frac{M}{2\sqrt{M^2-1}}(F_1 + F_2) \qquad \text{(VIII-17a)}$$

$$F_r = \frac{M}{2}(F_1 - F_2) \qquad \text{(VIII-17b)}$$

Now, with Eqs. (16) and (17), the basic equations are written,

$$\phi_{12} = \phi_{21} = \frac{1}{2rM}(\phi_1 - \phi_2) + (k^2 - \frac{n^2}{r^2M^2})\phi - i\frac{kM}{\sqrt{M^2-1}}(\phi_1 + \phi_2) \qquad \text{(VIII-18a)}$$

$$\phi_r = \frac{M}{2}(\phi_1 - \phi_2) = Z_x + ikZ \qquad \text{(VIII-18b)}$$

$$C_p = -2[\phi_x + ik\phi] \qquad \text{(VIII-18c)}$$

C. FINITE DIFFERENCE EQUATIONS

1. General Field Point

The finite difference equation for the points P and S along the right-running Mach line is,

$$\phi_2(S) - \phi_2(P) = \frac{\Delta s}{2rM}(\phi_1 - \phi_2) + \Delta s(k^2 - \frac{n^2}{M^2r^2})\phi - i\frac{kM\Delta s}{\sqrt{M^2-1}}(\phi_1 + \phi_2) \qquad \text{(VIII- 19)}$$

Note that in this and subsequent equations, $ds_1 = ds_2 = \Delta s$. For points Q and S along the left-running Mach line,

$$\phi_1(S) - \phi_1(Q) = \frac{\Delta s}{2rM}(\phi_1 - \phi_2) + \Delta s\left(k^2 - \frac{n^2}{M^2r^2}\right)\phi - i\frac{kM\Delta s}{\sqrt{M^2-1}}(\phi_1 + \phi_2) \qquad \text{(VIII- 20)}$$

The computational molecule is shown in figure 52.

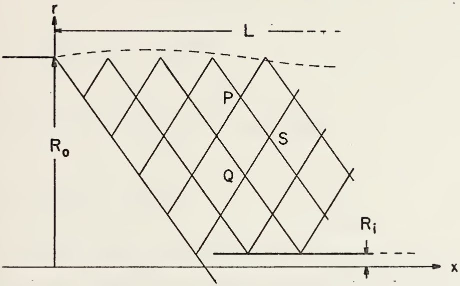


Figure 52. Characteristic grid with computational molecule.

The values for ϕ, ϕ_1 and ϕ_2 are averages between adjacent grid points along the respective Mach lines, so for the left-running Mach line,

$$\phi_1 = \frac{1}{2}[\phi_1(Q) + \phi_1(S)] \qquad \text{(VIII-21a)}$$

$$\phi_2 = \frac{1}{2}[\phi_2(Q) + \phi_2(S)] \qquad \text{(VIII-21b)}$$

and $\phi(S)$ is obtained by integrating along the Mach line.

$$\phi(S) = \phi(Q) + \frac{1}{2}[\phi_1(Q) + \phi_1(S)]\Delta s \qquad \text{(VIII-21c)}$$

$$\phi = \frac{1}{2}[\phi(S) + \phi(Q)] \qquad \text{(VIII-21d)}$$

Similarily, along the right-running Mach line,

$$\phi_1 = \frac{1}{2}[\phi_1(P) + \phi_1(S)] \qquad \text{(VIII-22a)}$$

$$\phi_2 = \frac{1}{2}[\phi_2(P) + \phi_2(S)] \qquad \text{(VIII-22b)}$$

$$\phi(S) = \phi(P) + \frac{1}{2}[\phi_2(P) + \phi_2(S)]\Delta s \qquad \text{(VIII-22c)}$$

$$\phi = \frac{1}{2}[\phi(P) + \phi(S)] \qquad \text{(VIII-22d)}$$

Eqs. (21) and (22) are substituted into Eqs. (19) and (20). The result is a system of two equations for $\phi_1(S)$ and $\phi_2(S)$ of the form,

$$A\phi_1(S) + B\phi_2(S) = C \qquad \text{(VIII-23a)}$$

$$D\phi_1(S) + E\phi_2(S) = F \qquad \text{(VIII-23b)}$$

Eq. (23a) applies to the right-running Mach line, where

$$A = 1 - \frac{\Delta s}{4Mr(S)} + i\frac{kM\Delta s}{2\sqrt{M^2-1}} \qquad \text{(VIII-24a)}$$

$$B = \frac{\Delta s}{4Mr(S)} - \left(k^2 - \frac{n^2}{M^2r^2(S)}\right)\left(\frac{\Delta s}{2}\right)^2 + i\frac{kM\Delta s}{2\sqrt{M^2-1}} \qquad \text{(VIII-24b)}$$

$$
\begin{aligned}
C = {} & \phi_2(P)\left[- \frac{\Delta s}{4Mr(P)} + \left(k^2 - \frac{n^2}{M^2 r^2(S)} \right)\left(\frac{\Delta s}{2}\right)^2 - i\frac{kM\Delta s}{2\sqrt{M^2-1}} \right] \\
& + \phi_1(P)\left[1 + \frac{\Delta s}{4Mr(P)} - i\frac{kM\Delta s}{2\sqrt{M^2-1}} \right] \\
& + \phi(P)\left[\left(k^2 - \frac{n^2}{M^2 r^2(P)} \right) + \left(k^2 - \frac{n^2}{M^2 r^2(S)} \right) \right]\left(\frac{\Delta s}{2}\right)
\end{aligned}
\tag{VIII-24c}
$$

Eq. (23b) applies to the left-running Mach line, with,

$$
D = \frac{\Delta s}{4Mr(S)} + \left(k^2 - \frac{n^2}{M^2 r^2(S)} \right)\left(\frac{\Delta s}{2}\right)^2 - i\frac{kM\Delta s}{2\sqrt{M^2-1}}
\tag{VIII-25a}
$$

$$
E = - 1 - \frac{\Delta s}{4Mr(S)} - i\frac{kM\Delta s}{2\sqrt{M^2-1}}
\tag{VIII-25b}
$$

$$
\begin{aligned}
F = {} & \phi_1(Q)\left[- \frac{\Delta s}{4Mr(Q)} - \left(k^2 - \frac{n^2}{M^2 r^2(S)} \right)\left(\frac{\Delta s}{2}\right)^2 + i\frac{kM\Delta s}{2\sqrt{M^2-1}} \right] \\
& + \phi_2(Q)\left[- 1 + \frac{\Delta s}{4Mr(Q)} + i\frac{kM\Delta s}{2\sqrt{M^2-1}} \right] \\
& - \phi(Q)\left[\left(k^2 - \frac{n^2}{M^2 r^2(Q)} \right) + \left(k^2 - \frac{n^2}{M^2 r^2(S)} \right) \right]\left(\frac{\Delta s}{2}\right)
\end{aligned}
\tag{VIII-25c}
$$

Eqs. (23) are used to compute ϕ and its derivatives at a general flow field point.

2. Initial Mach Line

Along the initial right-running Mach line, $\phi = \phi_2 = 0$, and Eq. (23a) reduces to,

$$
\phi_1(S) = C/A
\tag{VIII-26a}
$$

and C simplifies to,

$$C = \phi_1(P)[1 + \frac{\Delta s}{4Mr(P)} - i\frac{kM\Delta s}{2\sqrt{M^2-1}}] \qquad \text{(VIII-26b)}$$

3. Inner Surface Of Outer Cylinder

The flow tangency condition prescribes the normal velocity at $r = R_0$. Thus,

$$\phi_r\Big|_{r=R_0} = \frac{M}{2}[\phi_1(S) - \phi_2(S)] = Z_x + ikZ \qquad \text{(VIII-27a)}$$

For the initial point on the panel, $\phi_2 = 0$, and

$$\phi_1(S) = \frac{2}{M}[Z_x + ikZ]\Big|_{x=0} \qquad \text{(VIII-27b)}$$

Thereafter, the equations for the left-running Mach lines are used to solve for ϕ, ϕ_1 and ϕ_2. Hence,

$$\phi_1(S) = [F + \frac{2}{M}(Z_x + ikZ)E]/(D + E) \qquad \text{(VIII-28a)}$$

$$\phi_2(S) = [F - \frac{2}{M}(Z_x + ikZ)D]/(D + E) \qquad \text{(VIII-28b)}$$

$$C_p = -2[\phi_x + ik\phi] \qquad \text{(VIII-28c)}$$

Then, ϕ_1 and ϕ_2 are used to compute ϕ_x and C_p.

4. Boundary Condition at the Axis

The boundary condition at r = 0 was approximated by

assuming a very thin solid cylinder with radius $r = R_i$ along the axis, extending the length of the cylinder. The form of the boundary condition applied at $r = R_i$ is dependent on the circumferential mode number n. For even values of n,

$$\phi_r \Big|_{r=R_i} = 0 \qquad n \text{ even} \qquad \text{(VIII-29a)}$$

Thus,

$$\phi_1(S) = C/(A + B) = \phi_2(S) \qquad \text{(VIII-29b)}$$

For odd values of n, the prescribed boundary condition is $C_p = 0$. For the steady case ($k = 0$), this condition reduces to $\phi_x = 0$. Then applying the boundary condition with Eqs. (23a) and (17),

$$C_p\Big|_{r=R_i} \quad 0 = \phi_x(S) + ik\phi(S) \qquad n \text{ odd} \qquad \text{(VIII-30a)}$$

$$\phi(S) = \phi(P) + \frac{1}{2}[\phi_1(P) + \phi_2(S)]\Delta s \qquad \text{(VIII-30b)}$$

$$\phi_x(S) = \frac{M}{2\sqrt{M^2-1}}[\phi_1(S) + \phi_2(S)] = -ik\phi(S) \qquad \text{(VIII-30c)}$$

$$A\phi_1(S) + B\phi_2(S) = C \qquad \text{(VIII-30d)}$$

Defining,

$$\delta = \frac{M}{2\sqrt{M^2-1}} + ik\left(\frac{\Delta s}{2}\right) \qquad \text{(VIII-31a)}$$

$$\gamma = ik[\phi(P) + \left(\frac{\Delta s}{2}\right)\phi_2(P)] \qquad \text{(VIII-31b)}$$

Then,

$$\phi_1(S) = [\ C + B\frac{\gamma}{\delta}]/[\ A - \frac{M}{2\sqrt{M^2-1}}\frac{B}{\delta}\ ] \qquad \text{(VIII-31c)}$$

$$\phi_2(S) = [\ C - A\phi_1(S)\gamma\ ]/B \qquad \text{(VIII-31d)}$$

Now $\phi(S)$, $\phi_x(S)$, and $\phi_r(S)$ are computed using Eqs. (17) and (22a).

The initial right-running Mach line is assumed not to reflect from the assumed inner cylinder.

D. GENERALIZED FORCES AND THE PRESSURE COEFFICIENT

Having obtained the pressure distribution over the inner surface of the outer cylinder using the pressure coefficient amplitude, the generalized aerodynamic force Q_{mr} is found from,

$$Q_{mr} = \frac{1}{2}\int_0^1 C_{pm}(x)\,\psi_r(x)\,dx \qquad \text{(VIII-33a)}$$

where,

$$Z(x) = \sum_j A_j\psi_j(x) \qquad \text{(VIII-33b)}$$

and $C_{pm}(x)$ is the pressure distribution resulting from the m^{th} axial mode deflection and $\psi_r(x)$ is the r^{th} axial deflection mode. Thus, Q_{12} would be the generalized force due to an axial deflection of at least two modes of the

form,

$$Z(x) = A_1\psi_1(x) + A_2\psi_2(x) \qquad \text{(VIII-34a)}$$

and,

$$Q_{12} = \frac{1}{2}\int_0^1 C_{p_1}(x)\psi_2(x)\,dx \qquad \text{(VIII-34b)}$$

The above equations were programmed in Fortran IV and run on an IBM 360/67 computer. For 200 grid points along the flexible cylinder length, total run time was approximately twelve to fifteen seconds when using a precompiled source deck. The program is listed after the Two Degree of Freedom cascade listing.

E. RESULTS

Pressure coefficients and generalized aerodynamic forces were computed for sinusoidal deflection modes,

$$Z(x) = \sum_j \psi_j(x) = \sin(j\pi x) \;, \quad j = 1,2, \bullet \bullet \bullet \qquad \text{(VIII- 35)}$$

These cases were previously studied by Widnall and Dowell, [Ref. 27], who derived a solution of the full linearized equation using a Laplace-transform technique.

For figures 53 through 55 the grid fineness (number of increments along a Mach line between the inner and outer cylinder) was adjusted as a function of the difference in radii (or R/L) to maintain 200 points along the unit cylinder length. The inner radius was kept fixed at 0.0002 while the outer radius was varied from 0.1 to 5.0.

For low values of the circumferential mode number n, pressure distributions were excellent, with sharp discontinuities evident at reflection points on the outer cylinder. However, for n > 2, these reflection points exhibited a marked oscillation in pressure whose amplitude and duration were dependent on the inner radius and grid fineness used. By arbitrarily varying these input parameters, the oscillations could be minimized, but not entirely removed. Over a small range of inner radii, 0.01 to 0.0002, overall effect on Q_{mr} was relative small with differences of less than fifteen percent in most cases. For R/L large enough that no reflection occurred, Q_{mr} and pressure distributions were independent of inner radius size and grid fineness used. Further investigation is being conducted to develop stability criteria as a function of Mach number, grid fineness, and R/L.

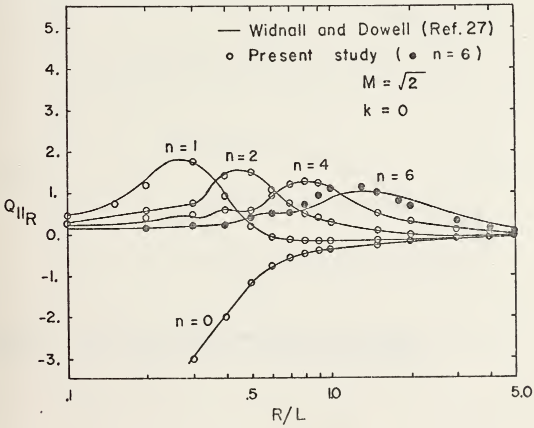


Figure 53. Generalized aerodynamic force Q_{11R} versus radius/length ratio as a function of the circumferential mode number (R_i=.0002).

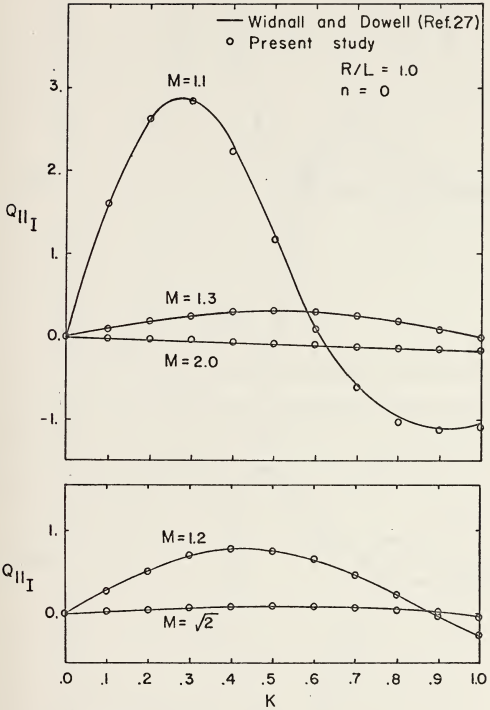


Figure 54. Generalized aerodynamic force Q_{11I} versus reduced frequency as a function of Mach number (R_i=.0002).

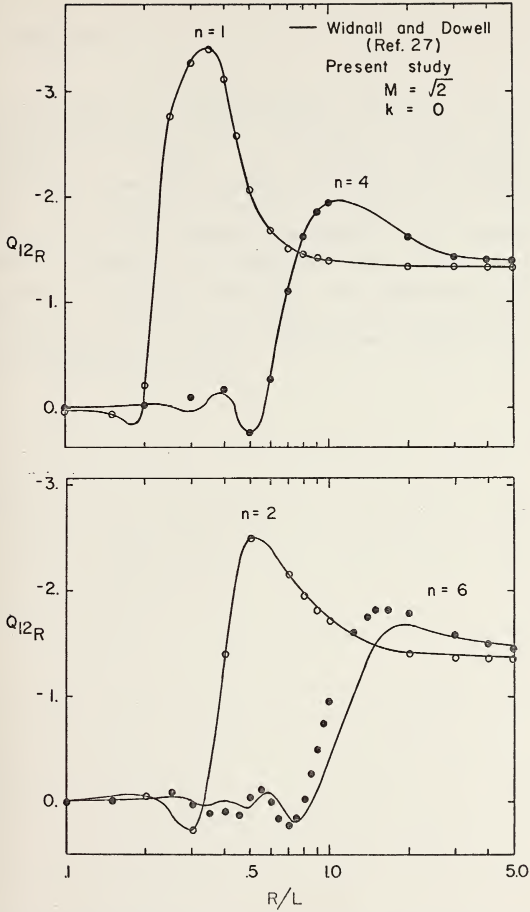


Figure 55. Generalized aerodynamic force Q_{12R} versus radius/length ratio as a function of the circumferential mode number (R_i=.0002).

F. CONCLUSIONS

The method of characteristics solution of the full linearized unsteady velocity potential equation has been developed to compute the pressure distributions and generalized forces for harmonically oscillating cylindrical shells with arbitrary radius-to-length ratios, axial and circumferential mode numbers, supersonic Mach numbers and reduced frequency. In addition, the method is easily adaptable to include arbitrary (non-sinusoidal) boundary conditions. Moreover, it is felt to be more easily applied than the complex integration techniques required for the Laplace-transform solution.

APPENDIX A

TWO DEGREE OF FREEDOM CASCADE

The following variables appear in the logic statements of the program.

ICO	used in subroutine GENFPT. When equal to one, it indicates that the previous grid point was a lower surface (blade or wake) point. ICO is always set equal to zero before exiting from GENFPT.
ID	integer array with each element corresponding to a cascade blade. Normally zero, elements are set equal to one the first time COEF2 is called for each blade.
IFLUT	an input variable. Set equal to one if flutter solutions are not desired; zero otherwise.
IHALT	a termination flag. When equal to one, X, as determined from the real or imaginary component of the flutter determinant, is negative.
IHAVEP	the number of the computation step along a right-running Mach line. Set equal to zero at the initial left-running Mach line, it is incremented until it is equal to LIMIT.
IPRES1,IPRES2	input variables identifying blades for which specific information is desired
ITER	iteration counter for the number of times through the flutter routine.
ITO	an information flag. If other than zero on return from FLUTER, the discriminant of the quadratic for the real roots of X is less than zero.
ITRMAX	maximum number of flutter solution iterations

	allowed before program termination.
IWRITE	output variable. Set equal to one in COEF2 after integration of the pressure distribution for the n^{th} blade is completed.
JLINE	the Mach line counter. Incremented each time a right-running Mach line is completed, it is reset to the value of IHAVEP at the leading edge of a new blade.
LB	an integer array. This is the first subscript of the matrix used for storing the discrete pressure and incremental moment distributions for each blade.
LCOUNT	the value of IHAVEP at a surface or wake grid point. When more than one blade is crossed by a right-running Mach line, LCOUNT is set equal to OLDL at the initial left-running Mach line and incremented by NGRDFN after each blade.
LIMIT	the maximum value of IHAVEP along a right-running Mach line.
LIMW	the value of LCOUNT at the last grid point on a blade. If IHAVEP is greater than LIMW, the grid point is in the wake.
LVEC	identical to LB, this value is passed to the main program via common.
NGRDFN	an input variable. It is the number of increments along a right-running Mach line between blades.
NEWDST	the number of Δx increments along a blade chord (of unit length).
NITOT	the total number of grid points on a blade. This variable is passed to the main program via common.
NN	an integer array in COEF used to identify successive grid points on a blade.

NSTPTS	the number of Δx increments between the initial Mach line and the leading edge of the second blade.
NUM	one less than the blade actually encountered. The program numbers blades from zero to NUMBLD - 1.
NUMBLD	the number of blades in the cascade.
NUMOLD	preserves the maximum value of NUM encountered along a right-running Mach line.
NUMPI	NUMOLD + 1.
MAXI	the value of JLINE on a right-running Mach line which intersects the leading edge of a blade.

Other program variables.

AI	A_I, the factor $\frac{1}{2}k\Delta x$
A1,A2	real and imaginary components of P_h
A3,A4	real and imaginary components of P_α
ANGP12,ANGM12	phase angle between real and imaginary components of the lift force and moment due to plunge
ANGP34,ANGM34	same as for ANGP12,ANGM12, but for pitch
BI	B_I, the factor $\frac{1}{4}k\frac{M^2}{M^2-1}\Delta x$
CR,CI	C_R,C_I, defined by Eqs. (IV-13c,d)
DELTA	the interblade phase angle in radians
DELTAS	the step size along a characteristic (Mach line)
DR,DI	D_R,D_I, defined by Eqs. (IV-13e,f)
DSTSTR	the step size along a blade
FAZE	the interblade phase angle in degrees
FF1,FF2,FF3	the flutter frequency based on the imaginary root and the two real roots respectively

FS1,FS2,FS3	the flutter speed based on the imaginary root and the two real roots respectively
FSTRMN	the input Mach number
G1, ... G9	variously defined in each subroutine which solves for perturbation quantities
HDSTRL	one-half DSTSTR
K12R, ... K56I	defined in III-B for each subroutine
L1, ... L4	real and imaginary lift coefficients for plunge and pitch oscillation respectively. Defined in Section IV.
M1, ... M4	real and imaginary moment coefficients for plunge and pitch oscillation respectively. Defined in Section IV
OMEGAA	the factor μr^2
OMEGAH	the factor $\mu(\frac{\omega_h}{\omega_\alpha})^2$
PR12,PI12	matrix storage for the real and imaginary pressure distribution resulting from plunge
PR34,PI34	matrix storage for the real and imaginary pressure distribution resulting from pitch
P12MAG,P34MAG	absolute magnitude of the complex lift force due to plunge and pitch rexpectively
REDFRQ	the reduced frequency
RINV	(1/k)
RMR12,RMI12	real and imaginary components of the moment distribution due to plunge
RMR34,RMI34	real and imaginary components of the moment distribution due to pitch
R12MAG,R34MAG	absolute magnitude of the complex moment resulting from plunge and pitch respectively
RL1,RL2	real and imaginary component of L_h
RL3,RL4	the real and imaginary components of L_α
RM1,RM2	real and imaginary components of M_h

RM3,RM4	real and imaginary components of M_α
RMUU	the blade density parameter
RSUBA	the radius of gyration
S	the factor $\sqrt{M^2-1}$
STGANG	the input stagger angle, in degrees
STEP	amount by which (1/k) is incremented after a call to FLUTER
TNWDST	input variable defining the vertical distance between adjacent blades
TRNGLH	sin($\bar{\alpha}$)*DELTAS. The distance between blades divided by the grid fineness ratio
U22R, ... C22I	perturbation quantities resulting from pitch at grid points not on a lower surface
U33R, ... C33I	same as above, but at grid points on a lower surface
U44R, ... C44I	perturbation quantities resulting from plunge at grid points not on a lower surface
U55R, ... C55I	same as above, but for grid points on a lower surface
VRPLNG,VIPLNG	real and imaginary components of the normal velocity perturbation due to plunge at the leading edge of the first blade
VRPANL,VIPANL	real and imaginary components of the normal velocity perturbation due to pitch at the leading edge of the first blade
W	the factor $\frac{M^2}{M^2-1}$
WHWA	the ratio of natural frequencies of plunge to pitch
XI	$\sqrt{\mathrm{Im}\{X\}}$
XN	floating point value of NUM
XNEW	blade position relative to the leading edge of the plade
XLNGTH	length of a Mach line between adjacent blades
XPT	matrix array to store the relative blade

	position of each grid point
XR1,XR2	the square roots of the two real roots of the flutter determinant
XSUBA	the position of the center of gravity, measured from the elastic axis
XSUBO	the elastic axis position

APPENDIX B

PANEL FLUTTER

The following variables appear in the Cylindrical Shell/Panel Flutter program.

A1	the factor $\frac{\Delta s}{4M}$
A2	the factor $\frac{M}{2\sqrt{M^2-1}}$
A3	the factor $\frac{kM\Delta s}{2\sqrt{M^2-1}}$
A4	the factor $\frac{M}{2}$
BETA	$\sqrt{M^2-1}$
CPR,CPI	real and imaginary components of the pressure coefficient
DELTAS	distance along a Mach line between radii
DPHIS1	the directional derivative of ϕ along the left-running Mach line
DPHIS2	the directional derivative of ϕ along the right-running Mach line
DPHIDX	ϕ_x
DPHIDR	ϕ_r
FACT1	k^2
FACT2	$(\frac{n}{M})^2$
FSTRMN	the Mach number
IHAVEP	the number of a computational step along a right-running Mach line
IPRINT	set equal to one if the user desires the entire flow field printed out. Otherwise, zero
LCOUNT	consecutive number of the grid points on the

	outer cylinder surface. Incremented by one after each call to RAD1
M	the axial mode number, m
M1	mode number of the panel displacement used in the computation of Q_{mr}
N	the circumferential mode number, n
NGRDFN	the number of increments along a right-running Mach line between the outer and inner cylinder
QREAL,QIMAG	real and imaginary components of Q_{mr}
REDFRQ	the reduced frequency k
RI,RO	the inner and outer cylinder radii
TANR	the flow tangency condition
XPT	axial position of grid points on the flexible panel surface

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                          C
C             TWO DEGREE OF FREEDOM PROGRAM                C
C                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C                THIS IS THE MAIN PROGRAM
C
C
C
C                    *****  NOTE  *****
C
C     THE SIZE OF THE ARRAYS IN THIS PROGRAM ARE SUBJECT
C     TO THE FOLLOWING CONSTRAINT:
C           NGRDFN*TAN(A)*( (N-1)*(TAN(B) - COT(A) ) + 1/D )
C               LESS THAN TWICE THE DIMENSION OF THE ARRAY.
C                      A  -  MACH ANGLE
C                      B  -  STAGGER ANGLE
C                      D  -  DISTANCE BETWEEN BLADES
C                      N  -  NUMBER OF BLADES
C
      REAL*8 X,Y,DSTSTR,HDSTRL,TRNGLH,STAG,FSTRMN
      INTEGER OLOL
      DIMENSION X(400),Y(400)
      DIMENSICN U22R(400),U22I(400),V22R(400),V22I(400),
     1   C22R(400),C22I(400)
      DIMENSION U33R(400),U33I(400),V33R(400),V33I(400),
     1   C33R(400),C33I(400)
      DIMENSION U44R(400),U44I(400),V44R(400),V44I(400),
     1   C44R(400),C44I(400)
      DIMENSION U55R(400),U55I(400),V55R(400),V55I(400),
     1   C55R(400),C55I(400)
      DIMENSION RL1(76),RL2(76),RL3(76),RL4(76),RM1(76),
     1   RM2(76),RM3(76),RM4(76),ANGP12(76),ANGM12(76),
     2   ANGP34(76),ANGM34(76),P12MAG(76),P34MAG(76),
     3   R12MAG(76),R34MAG(76)
      DIMENSION XPT(4,100),PR12(4,100),PI12(4,100),PR34(4,
     1   100),PI34(4,100)
      COMMON /BLCK1/ X,Y,REDFRQ,VRPANL,VIPANL,VRPLNG,VIPLNG,
     1   S,W,AI,BI
      CCMMON /BLCK2/ U22R,U22I,V22R,V22I,C22R,C22I,
     1   U2RNEW,U2INEW,V2RNEW,V2INEW,C2RNEW,C2INEW
      COMMON /BLCK3/ U33R,U33I,V33R,V33I,C33R,C33I,
     1   U3RNEW,U3INEW,V3RNEW,V3INEW,C3RNEW,C3INEW
      COMMON /BLCK4/ IHAVEP,NGRDFN,NUM,NUMBLD,NSTPTS,LCOUNT,
     1   JLINE,ICO,IPRES1,IPRES2
      COMMON /BLCK5/ U44R,U44I,V44R,V44I,C44R,C44I,
     1   U4RNEW,U4INEW,V4RNEW,V4INEW,C4RNEW,C4INEW
      COMMON /BLCK6/ U55R,U55I,V55R,V55I,C55R,C55I,
     1   U5RNEW,U5INEW,V5RNEW,V5INEW,C5RNEW,C5INEW
      COMMON/BLCK7/ RL1,RL2,RL3,RL4,RM1,RM2,RM3,RM4,ANGP12,
     1   ANGM12,ANGP34,ANGM34,P12MAG,R12MAG,P34MAG,R34MAG
      COMMCN/BLCK8/ XPT,PR12,PI12,PR34,PI34,IWRITE,LVEC,
     1   NITOT,NJUNK
      COMMON /GEOM/ FSTRMN,DSTSTR,HDSTRL,TRNGLH,STAG,XSUBO,
     1   DELTA,FAZE
      COMMON/FLUTR/ RMUU,XSUBA,RSUBA,WHWA,STEP,IFLUT
C
C     PRINT OUT ALL INPUT INFORMATION VIA INPUT
C
C
      WRITE(6,16)
C
      CALL INPUT(IHALT)
C
C     IF IHALT EQUAL 1, TERMINATE. A DIMENSION SIZE HAS BEEN
C     EXCEEDED.
C
      IF ( IHALT .EQ. 1 ) GO TO 14
C
C     THE OUTER LOOP IS USED FOR FLUTTER ITERATION.
C
```

```
    1 CONTINUE
C
C     INITIALIZE FLOW FIELD QUANTITIES AND LOGIC VARIABLES
C
      CALL INITAL
      CALL COEF
      FAC = 1./DSTSTR
      NEWDST = FAC
      LIMIT = ( NUMBLD - 1 )*NSTPTS + NEWDST + 1
      LIMW = NEWDST
      MAXI = NGRDFN + NSTPTS
      NUM = 0
      NUMOLD = 0
      NUMPI = 1
      OLDL = 0
      ICO = 0
      IWRITE = 0
C
C     THE INNER LOOP IS FOR FLOW FIELD CALCULATION
    2 CONTINUE
C
C
      CALL COMPXY
C
C     IF POINT IS ON THE INITIAL RIGHT RUNNING MACH LINE,
C     CALL MACHLN.
C
      IF ( LCOUNT .EQ. 0 ) GO TO 3
C
C     IF THE POINT IS ON A BLADE OR IN A WAKE, GO TO 4
C
      IF ( IHAVEP .EQ. LCOUNT ) GO TO 4
C
C     IF THE POINT IS ON THE INITIAL LEFT RUNNING MACH LINE,
C     CALL MACHLN.
C
      IF ( IHAVEP .EQ. 0 ) GO TO 3
C
C     CONDITION FOR THE FIRST ENCOUNTER WITH A BLADE
C
      IF (( IHAVEP .EQ. NUMPI*NSTPTS ) .AND.
     1   ( JLINE .EQ. MAXI )) GO TO 10
C
      CALL GENFPT
      IF ( IHAVEP .EQ. LIMIT ) GO TO 7
      IHAVEP = IHAVEP + 1
      GO TO 2
C
    3 CALL MACHLN
      IF ( IHAVEP .EQ. LIMIT ) GO TO 7
      IHAVEP = IHAVEP + 1
      GO TO 2
C
    4 ICO = 1
C
C     IF THE POINT IS IN THE WAKE, CALL WAKE
C
      IF ( IHAVEP .GT. LIMW ) GO TO 5
      CALL TOP
      CALL BOTTOM
      CALL COEF1
      IHAVEP = IHAVEP + 1
      IF ( NUM .EQ. 0 ) GO TO 2
      NUM = NUM - 1
      LCOUNT = LCOUNT + NGRDFN
      LIMW = LIMW - NSTPTS
      GO TO 2
C
    5 CALL WAKE
      CALL COEF2
C
C     IF IWRITE EQUAL  1  ON RETURN FROM COEF2, IS THE
```

```
C     PRESSURE DISTRIBUTION FOR THIS BLADE DESIRED?
C
      IF ( IWRITE .NE. 1 ) GO TO 6
      M = NUM + 1
      IF ( ( M .EQ. IPRES1 ) .OR. ( M .EQ. IPRES2 ) )
     1             GO TO 14
    6 IWRITE = 0
      IF ( IHAVEP .EQ. LIMIT ) ICO = 0
C
C     IF THIS IS THE LAST POINT TO BE COMPUTED IN THE FLOW
C     FIELD AREA, TERMINATE.
C
      IF (( NUM .GE. NUMBLD-1 ).AND.(IHAVEP .EQ. LIMIT ))
     1             GO TO 11
C
C     IF THIS IS THE LAST GRID POINT ON A RIGHT-RUNNING
C     MACH LINE, SWITCH THE THE LOGIC VARIABLES AND GO TO
C     THE NEXT MACH LINE. OTHERWISE, CONTINUE.
C
      IF ( IHAVEP .EQ. LIMIT ) GO TO 7
      IHAVEP = IHAVEP + 1
      IF ( NUM .EQ. 0 ) GO TO 2
      LCOUNT = LCOUNT + NGRDFN
      NUM = NUM - 1
      GO TO 2
C
    7 CONTINUE
C
C     IS THIS THE INITIAL RIGHT-RUNNING MACH LINE?
C
      IF ( LCOUNT .EQ. 0 ) GO TO 8
C
C     THIS IS THE LAST GRID POINT ON A RIGHT-RUNNING MACH
C     LINE. STORE THE PERTURBATION QUANTITIES JUST COMPUTED
C     AND GO TO THE NEXT MACH LINE.
C
      I = IHAVEP + 1
      U22R(I) = U2RNEW
      U22I(I) = U2INEW
      V22R(I) = V2RNEW
      V22I(I) = V2INEW
      C22R(I) = C2RNEW
      C22I(I) = C2INEW
C
      U44R(I) = U4RNEW
      U44I(I) = U4INEW
      V44R(I) = V4RNEW
      V44I(I) = V4INEW
      C44R(I) = C4RNEW
      C44I(I) = C4INEW
C
C     SWITCH THE LOGIC VARIABLES TO START THE COMPUTATION.
C
    8 IHAVEP = 0
      NUM = NUMOLD
      ICO = 0
      JLINE = JLINE + 1
C
C     IF THIS IS NOT THE FIRST BLADE, RESET THE LOGIC SWITCH
C     VARIABLES LCOUNT, LIMW, OLDL.
C
      IF ( NUMOLD .GT. 0 ) GO TO 9
      LCOUNT = LCOUNT + 1
      GO TO 2
C
    9 LCOUNT = OLDL
      LIMW = NUMOLD*NSTPTS + NEWDST
      OLDL = OLDL + 1
      GO TO 2
C
   10 CALL NEWBLD
      ICO = 1
```

```
      CALL COEF1
      NUMOLD = NUM
      LCOUNT = IHAVEP + NGRDFN
      JLINE = IHAVEP
      IHAVEP = IHAVEP + 1
      OLDL = IHAVEP
      NUMPI = NUM + 1
      MAXI = NGRDFN + NUMPI*NSTPTS
      NUM = NUM - 1
      GO TO 2
C
   11 CONTINUE
C
      IF ( IFLUT .EQ. 1 ) GO TO 14
      CALL FLUTER(REDFRQ,NUMBLD,ITO,IHALT)
C
      IF ( IHALT .EQ. 1 ) GO TO 14
      RINV = 1./REDFRQ
      IF ( ITO .EQ. 1 ) GO TO 13
      RINV = RINV + STEP
      REDFRQ = 1./RINV
      GO TO 1
   12 RINV = RINV - STEP
      STEP = STEP/5.
      WRITE(6,17) STEP
      RINV = RINV + STEP
      REDFRQ = 1./RINV
      GO TO 1
   13 CONTINUE
C
   14 CONTINUE
      WRITE(6,18)
      DO 15 N = 1,NUMBLD
   15 WRITE(6,19) N,P12MAG(N),ANGP12(N),R12MAG(N),
     1   ANGM12(N),P34MAG(N),ANGP34(N),R34MAG(N),
     2   ANGM34(N)
      N = 14
      WRITE(6,100)FAZE,RL1(N),RL2(N),RM1(N),RM2(N),RL3(N),
     1   RL4(N),RM3(N),RM4(N)
  100 FORMAT('0',5X,9F12.4)
      WRITE(6,16)
      STOP
   16 FORMAT('1')
   17 FORMAT('0',35X,'***  NEW STEP = ',F6.4,'  ***')
   18 FORMAT('0',5X,'BLADE    P12MAG    ANGP12    R12MAG',
     1  '    ANGM12    P34MAG    ANGP34    R34MAG    ',
     2  'ANGM34',//)
   19 FORMAT(' ',4X,I5,1X,8F10.4,/)
      END



C                 THIS IS SUBROUTINE FLUTER
C
      SUBROUTINE FLUTER(REDFRQ,NUMBLD,ITO,IHALT)
C
      REAL L1,L2,L3,L4,M1,M2,M3,M4
      DIMENSION RL1(76),RL2(76),RL3(76),RL4(76),RM1(76),
     1   RM2(76),RM3(76),RM4(76),ANGP12(76),ANGM12(76),
     2   ANGP34(76),ANGM34(76),P12MAG(76),P34MAG(76),
     3   R12MAG(76),R34MAG(76)
      COMMON/BLCK7/ RL1,RL2,RL3,RL4,RM1,RM2,RM3,RM4,ANGP12,
     1   ANGM12,ANGP34,ANGM34,P12MAG,R12MAG,P34MAG,R34MAG
      COMMON/FLUTR/ RMUU,XSUBA,RSUBA,WHWA,STEP,IFLUT
      DATA IPRINT /1/
C
      IF ( IPRINT .EQ. 2 ) GO TO 1
      IPRINT = 2
      WRITE(6,115)
      WRITE(6,116)
    1 CONTINUE
```

```
      ITO = 0
      II = NUMBLD
C
C     FACTOR IS USED TO MAKE LIFT AND MOMENT VALUES AGREE
C     WITH GARRICK AND RUBINOW NOTATION.
C
      FACTOR = 0.5/(REDFRQ*REDFRQ)
C
      L1 = RL1(II)*FACTOR
      L2 = RL2(II)*FACTOR
      L3 = RL3(II)*2.*FACTOR
      L4 = RL4(II)*2.*FACTOR
      M1 = RM1(II)*2.*FACTOR
      M2 = RM2(II)*2.*FACTOR
      M3 = RM3(II)*4.*FACTOR
      M4 = RM4(II)*4.*FACTOR
C
C     DEFINE CONSTANTS TO BE USED IN THE FLUTER SOLUTION.
C
      RINV = 1./REDFRQ
      W2 = WHWA*WHWA
      R2 = RSUBA*RSUBA
      X2 = XSUBA*XSUBA
      OMEGAA = RMUU*R2
      OMEGAH = RMUU*W2
C
      DI = L1*M4 - L4*M1 + L2*M3 - L3*M2
      DR = L1*M3 - L3*M1 - L2*M4 + M2*L4
C
      CR = RMUU*( XSUBA*( M1 + L3 ) - ( M3 - OMEGAA ) -
     1   ( L1*R2 + RMUU*X2 ) ) + DR
      CI = RMUU*( XSUBA*( M2 + L4 ) - M4 - L2*R2 ) + DI
C
      IF ( WHWA .EQ. 0. ) GO TO 20
C
C     DEFINE CONSTANTS FOR QUADRATIC EQUATION OF REAL
C     COMPONENT AND FOR THE IMAGINARY COMPONENT.
C
      B = (L1 - RMUU )/OMEGAH + M3/OMEGAA - 1.
      C = CR/(OMEGAA*OMEGAH)
      D = L2*OMEGAA + OMEGAH*M4
C
C     SOLVE FOR AND TEST THE DISCRIMINANT
C
      DISCRM = B*B - 4.*C
      IF ( DISCRM .LT. 0. ) GO TO 40
C
C     SOLUTION OF IMAGINARY COMPONENT
C
      XII = - CI/D
      IF ( XII .LT. 0. ) GO TO 41
      XI = SQRT(XII)
C
C     SOLUTION OF REAL COMPONENT
C
      XR11 = ( - B + SQRT(DISCRM) )*0.5
      XR21 = ( - B - SQRT(DISCRM) )*0.5
      IF ( ( XR11 .LT. 0. ) .OR. ( XR21 .LT. 0. ) ) GO TO 42
      XR1 = SQRT(XR11)
      XR2 = SQRT(XR21)
C
C     CALCULATION OF FLUTTER FREQUENCY (FF) AND SPEED (FS)
C     BASED ON ALL THREE ROOTS.
C
      FF1 = 1./XI
      FS1 = FF1*RINV
      FF2 = 1./XR1
      FS2 = FF2*RINV
      FF3 = 1./XR2
      FS3 = FF3*RINV
      WRITE(6,100) REDFRQ,RINV,XI,XR1,XR2,FF1,FS1,FF2,FS2,
     1   FF3,FS3
```

```
      RETURN
C
C     SOLUTION WHEN RATIO OF NATURAL FREQUENCIES IS ZERO
C
   20 F = OMEGAA*(L1 - RMUU)
C
C     SOLUTION FOR IMAGINARY ROOT
C
      XII = - CI/(OMEGAA*L2)
      IF ( XII .LT. 0. ) GO TO 41
      XI = SQRT(XII)
C
C     SCLUTION FOR REAL ROOT
C
      XR11 = - CR/F
      IF ( XR11 .LT. 0. ) GO TO 42
      XR1 = SQRT(XR11)
C
C     CALCULATION OF FLUTTER FREQUENCY (FF) AND SPEED (FS)
C     BASED ON THE REAL AND IMAGINARY COMPONENTS.
C
      FF1 = 1./XI
      FS1 = FF1*RINV
      FF2 = 1./XR1
      FS2 = FF2*RINV
      WRITE(6,117) REDFRQ,RINV,XI,XR1,FF1,FS1,FF2,FS2
C
      RETURN
   40 WRITE(6,110) REDFRQ,RINV
      ITO = 1
      RETURN
   41 WRITE(6,111) REDFRQ,RINV
      IHALT = 1
      RETURN
   42 WRITE(6,112) REDFRQ,RINV
      IHALT = 1
      RETURN
  100 FORMAT('0',5X,5F10.4,'   **',6F10.4)
  110 FORMAT('0', 5X,2F10.4,5X,'DISCRM < 0')
  111 FORMAT('0', 5X,2F10.4,15X,'XIMAG < 0',//)
  112 FORMAT('0', 5X,2F10.4,15X,'XREAL < 0',//)
  115 FORMAT('0',20X,'FLUTTER POINT IS THE INTERSECTION',
     1   ' OF THE REAL AND IMAGINARY ',/' ',20X,'CURVES ',
     2   ' OF SQRT(X) WHEN PLOTTED VERSUS (1/K).  AT THIS ',
     3   'POINT:',/' ',30X,'FLUTTER FREQUENCY (FF) = ',
     4  '1/SQRT(X)',/' ','FLUTTER SPEED (FS) = FF/K',/)
  116 FORMAT(' ',30X,'SQRT',5X,'SQRT',6X,'SQRT',12X,'BASED',
     1  ' ON XIMAG',5X,'BASED ON XREAL(+)',3X,'BASED ON ',
     2   'XREAL(-)',/' ',11X,'K',8X,'1/K',7X,'(XI)',5X,'(XR+
     3   )',5X,'(XR-)',12X,'FF',8X,'FS',8X,'FF',8X,'FS',8X,
     4   'FF',8X,'FS',//)
  117 FORMAT('0',5X,4F10.4,10X,'   **',4F10.4)
      END


C                   THIS IS SUBROUTINE INPUT
C
      SUBROUTINE INPUT(IHALT)
C
C     SUBROUTINE INPUT READS ALL GEOMETRIC AND AERODYNAMIC
C     INPUT DATA.
C
      REAL*8 X,Y,DSTSTR,HDSTRL,TRNGLH,STAG,FSTRMN,FMANGL,
     1   TNWDST,DELTAS,XLNGTH,STGR,SOLID,STGANG,ARG,FACTOR,
     2   ANG
      DIMENSION X(400),Y(400)
      COMMON /BLCK1/ X,Y,REDFRQ,VRPANL,VIPANL,VRPLNG,VIPLNG,
     1   S,W,AI,BI
      COMMON /BLCK4/ IHAVEP,NGRDFN,NUM,NUMBLD,NSTPTS,LCOUNT,
     1   JLINE,ICO,IPRES1,IPRES2
```

```
      COMMON /GEOM/ FSTRMN,DSTSTR,HDSTRL,TRNGLH,STAG,XSUBO,
     1   DELTA,FAZE
      COMMON/FLUTR/ RMUU,XSUBA,RSUBA,WHWA,STEP,IFLUT
      NAMELIST /NAM1/ NGRDFN,FSTRMN,REDFRQ,XSUBO,TNWDST,
     1   STGANG,FAZE,NUMBLD,IPRES1,IPRES2
      NAMELIST/FLTR/ RMUU,XSUBA,RSUBA,WHWA,STEP,IFLUT
      DATA MAXDIM/400/, MAXPTS/100/, MAXBLD/75/, ICROSS/4/
C
      READ(5,NAM1)
      WRITE(6,NAM1)
      READ(5,FLTR)
      WRITE(6,FLTR)
C
      FMANGL = DARSIN(1.D0/FSTRMN)
      XLNGTH = TNWDST*FSTRMN
      FNGDPT = DFLOAT(NGRDFN)
      DELTAS = XLNGTH/FNGDPT
      TRNGLH = TNWDST/FNGDPT
      HDSTRL = DELTAS*DCOS(FMANGL)
      DSTSTR = HDSTRL + HDSTRL
C
C     COMPUTATION OF THE COMPATIBLE STAGGER ANGLE. THIS IS
C     REQUIRED  BECAUSE THE LEADING EDGE OF EACH BLADE MUST
C     BE ON A GRID POINT.
C
      ARG = STGANG*1.74532925D-02
      STAG = TNWDST*DTAN(ARG)
      FACTOR = TNWDST/DTAN(FMANGL)
      STGR = STAG - FACTOR
      NSTPTS = STGR/DSTSTR
      IF ( NSTPTS .LT. 1 ) NSTPTS = 1
      STGR = DFLOAT(NSTPTS)*DSTSTR
      STAG = STGR + FACTOR
      ANG = DATAN( STAG/TNWDST )
      STGANG = ANG*57.2957795 1D0
C
      SOLID = DCOS(ANG)/TNWDST
      S = DSQRT( FSTRMN*FSTRMN - 1.D0 )
      W = FSTRMN*FSTRMN/(S*S)
      DELTA = FAZE*0.1745329E-01
C
      WRITE(6,10)NGRDFN,FSTRMN,REDFRQ,TNWDST,SOLID,XSUBO,
     1   STGANG,DSTSTR,FAZE,NUMBLD
C
C     CHECK WHETHER OR NOT VECTOR DIMENSIONS ARE EXCEEDED.
C     IBLDPT IS THE TOTAL NUMBER OF GRID POINTS ON A BLADE,
C     INTXN .LE. (IBLDPT+1) INDICATES THAT A MACHLINE
C     CROSSES TOO MANY BLADES, AND IPTS IS THE TOTAL NUMBER
C     OF GRID POINTS ALONG THE MACHLINES.  ICROSS IS THE
C     MAXIMUM NUMBER OF BLADES THAT CAN BE CROSSED BY A
C     RIGHT-RUNNING MACH LINE.
C
      IHALT = 0
C
      INTXN = ICROSS*( NSTPTS + NGRDFN )
      FAC = 1./DSTSTR
      IBLDPT = FAC + 2
      IPTS = ( NUMBLD - 1 )*NSTPTS + IBLDPT
C
      IF ( IPTS .GT. MAXDIM ) GO TO 4
    1 IF ( IBLDPT .GT. MAXPTS ) GO TO 5
    2 IF ( NUMBLD .GT. MAXBLD ) GO TO 6
    3 IF ( INTXN .LE. IBLDPT + 1 ) GO TO 7
      IF ( IHALT .EQ. 1 ) RETURN
      GO TO 8
C
    4 WRITE(6,120) MAXDIM
      IHALT = 1
      GO TO 1
C
    5 WRITE(6,121) MAXPTS
      IHALT = 1
```

```
      GO TO 2
C
    6 WRITE(6,122) MAXBLD
      IHALT = 1
      GO TO 3
    7 WRITE(6,123) ICROSS
      IHALT = 1
      RETURN
C
    8 CONTINUE
      RETURN
   10 FORMAT(////' ',5X,'GRID FINENESS INPUT NUMBER = ',I10,
     1   /'0',5X,'FREESTREAM MACH NUMBER = ',F10.7,/'0',5X,
     2   'REDUCED FREQUENCY = ',F10.7,/'0',5X,'DISTANCE ',
     3   'BETWEEN AIRFOILS = ',F10.7,/'0',5X,'SOLIDITY = ',
     4   F10.7,/'0',5X,'HORIXONTAL POSITION OF ELASTIC ',
     5   'AXIS = ',F10.7,/'0',5X,'COMPATABLE STAGGER ANGLE',
     6   ' = ',F10.7,/'0',5X,'DELTA X = ',F10.7,/'0',5X,
     7   'UPPER AIRLOIL LEADS LOWER BY ',F10.4,/'0',I3,
     8   '    BLADES',///)
   11 FORMAT('1',34X,'TEIPEL COMPLEX PERTURBATION ',
     1   'AMPLITUDES'///' ',1X,'BLD',T10,'X',T26,'UR',T42,
     2   'UI',T58,'VR',T73,'VI',T90,'CR',T107,'CI',/)
  120 FORMAT('0',2X,74('*'),/'0',5X,'ERROR TERMINATION: ',
     1   'NUMBER OF GRID POINTS ALONG A MACH LINE EXCEEDS ',
     2   I5,/'0',2X,74('*'))
  121 FORMAT('0',2X,74('*'),/'0',5X,'ERROR TERMINATION: ',
     1   'NUMBER OF POINTS ALONG A BLADE EXCEEDS ',I5,/'0',
     2   2X,74('*'))
  122 FORMAT('0',2X,54('*'), /'0',5X,'ERROR TERMINATION: ',
     1   'NUMBER OF BLADES EXCEEDS ',I4,/'0',2X,54('*'))
  123 FORMAT('0',2X,77('*'),/'0',5X,'ERROR TERMINATION: ',
     1   'RIGHT-RUNNING MACH LINE CROSSES MORE THAN ',I2,
     2   '  BLADES',/'0',2X,77('*'))
      END


C                   THIS IS SUBROUTINE INITAL
C
      SUBROUTINE INITAL
C
C     SUBROUTINE INITAL INITIALIZES ALL FLOW FIELD
C     QUANTITIES.
C
      REAL*8 X,Y,DSTSTR,HDSTRL,TRNGLH,STAG,FSTRMN
      DIMENSION X(400),Y(400)
      DIMENSION U22R(400),U22I(400),V22R(400),V22I(400),
     1   C22R(400),C22I(400)
      DIMENSION U33R(400),U33I(400),V33R(400),V33I(400),
     1   C33R(400),C33I(400)
      DIMENSION U44R(400),U44I(400),V44R(400),V44I(400),
     1   C44R(400),C44I(400)
      DIMENSION U55R(400),U55I(400),V55R(400),V55I(400),
     1   C55R(400),C55I(400)
      COMMON /BLCK1/ X,Y,REDFRQ,VRPANL,VIPANL,VRPLNG,VIPLNG,
     1   S,W,AI,BI
      COMMON /BLCK2/ U22R,U22I,V22R,V22I,C22R,C22I,
     1   U2RNEW,U2INEW,V2RNEW,V2INEW,C2RNEW,C2INEW
      COMMON /BLCK3/ U33R,U33I,V33R,V33I,C33R,C33I,
     1   U3RNEW,U3INEW,V3RNEW,V3INEW,C3RNEW,C3INEW
      COMMON /BLCK4/ IHAVEP,NGRDFN,NUM,NUMBLD,NSTPTS,LCOUNT,
     1   JLINE,ICO,IPRES1,IPRES2
      COMMON /BLCK5/ U44R,U44I,V44R,V44I,C44R,C44I,
     1   U4RNEW,U4INEW,V4RNEW,V4INEW,C4RNEW,C4INEW
      COMMON /BLCK6/ U55R,U55I,V55R,V55I,C55R,C55I,
     1   U5RNEW,U5INEW,V5RNEW,V5INEW,C5RNEW,C5INEW
      COMMON /GEOM/ FSTRMN,DSTSTR,HDSTRL,TRNGLH,STAG,XSUBO,
     1   DELTA,FAZE
C
      AI = .5*REDFRQ*SNGL(DSTSTR)
```

```
      BI = .5*AI*W
      X(1) = 0.0D0
      XI = X(1)
      Y(1) = 0.0D0
C
C     SET UP INITIAL VALUES FOR U,V, AND C AT (0,0)
C     PITCH:
C
      VRPANL = -1./S
      VIPANL = REDFRQ*XSUBO/S
C
C     PLUNGE:
C
      VRPLNG = 0.0
      VIPLNG = -REDFRQ/S
C
C     INITIAL VALUES OF PERTURBATION QUANTITIES  RESULTING
C     FROM PITCH OSCILLATION AT THE LEADING EDGE OF THE
C     FIRST BLADE; UPPER SURFACE (22), LOWER SURFACE (33)
C
      U22R(1) = -VRPANL
      U22I(1) = -VIPANL
      V22R(1) = -U22R(1)
      V22I(1) = -U22I(1)
      C22R(1) = -U22R(1)
      C22I(1) = -U22I(1)
      U33R(1) = VRPANL
      U33I(1) = VIPANL
      V33R(1) = VRPANL
      V33I(1) = VIPANL
      C33R(1) = -VRPANL
      C33I(1) = -VIPANL
C
C     INITIAL VALUES OF PERTURBATION QUANTITIES  RESULTING
C     FROM PLUNGE OSCILLATION AT THE LEADING EDGE OF THE
C     FIRST BLADE; UPPER SURFACE (44), LOWER SURFACE (55)
C
      U44R(1) = -VRPLNG
      U44I(1) = -VIPLNG
      V44R(1) = -U44R(1)
      V44I(1) = -U44I(1)
      C44R(1) = -U44R(1)
      C44I(1) = -U44I(1)
      U55R(1) = VRPLNG
      U55I(1) = VIPLNG
      V55R(1) = VRPLNG
      V55I(1) = VIPLNG
      C55R(1) = -VRPLNG
      C55I(1) = -VIPLNG
C
      JLINE = 0
      IHAVEP = 1
      LCOUNT = 0
      RETURN
  100 FORMAT(2H I,7E16.7,/)
      END


C                   THIS IS SUBROUTINE COMPXY
C
      SUBROUTINE COMPXY
C
C     COMPXY COMPUTES THE (X,Y) GRID POINT POSITION
C
      REAL*8 X,Y,DSTSTR,HDSTRL,TRNGLH,STAG,FSTRMN
      DIMENSION X(400),Y(400)
      CCMMON /BLCK1/ X,Y,REDFRQ,VRPANL,VIPANL,VRPLNG,VIPLNG,
     1   S,W,AI,BI
      COMMON /BLCK4/ IHAVEP,NGRDFN,NUM,NUMBLD,NSTPTS,LCOUNT,
     1   JLINE,ICO,IPRES1,IPRES2
```

```
      CCMMON /GEOM/ FSTRMN,DSTSTR,HDSTRL,TRNGLH,STAG,XSUBO,
     1   DELTA,FAZE
C
      I = IHAVEP + 1
C
C     IF GRID POINT IS ON THE INITIAL LEFT RUNNING MACH
C     LINE, GO TO 1.
C
      IF (( IHAVEP .EQ. 0 ).AND.( LCOUNT .NE. 0 )) GO TO 1
      X(I) = X(IHAVEP) + HDSTRL
      Y(I) = Y(IHAVEP) - TRNGLH
      RETURN
C
    1 X(I) = X(1) + HDSTRL
      Y(I) = Y(1) + TRNGLH
      RETURN
      END


C                 THIS IS SUBROUTINE MACHLN
C
      SUBROUTINE MACHLN
C
C     MACHLN COMPUTES THE VALUE OF U,V, AND C ALONG THE
C     INITIAL MACH LINE AT THE GIVEN GRID POINT.
C
      REAL*8 X,Y
      DIMENSION X(400),Y(400)
      DIMENSION U22R(400),U22I(400),V22R(400),V22I(400),
     1   C22R(400),C22I(400)
      DIMENSION U33R(400),U33I(400),V33R(400),V33I(400),
     1   C33R(400),C33I(400)
      DIMENSION U44R(400),U44I(400),V44R(400),V44I(400),
     1   C44R(400),C44I(400)
      DIMENSION U55R(400),U55I(400),V55R(400),V55I(400),
     1   C55R(400),C55I(400)
      CCMMON /BLCK1/ X,Y,REDFRQ,VRPANL,VIPANL,VRPLNG,VIPLNG,
     1   S,W,AI,BI
      COMMON /BLCK2/ U22R,U22I,V22R,V22I,C22R,C22I,
     1   U2RNEW,U2INEW,V2RNEW,V2INEW,C2RNEW,C2INEW
      COMMON /BLCK3/ U33R,U33I,V33R,V33I,C33R,C33I,
     1   U3RNEW,U3INEW,V3RNEW,V3INEW,C3RNEW,C3INEW
      COMMON /BLCK4/ IHAVEP,NGRDFN,NUM,NUMBLD,NSTPTS,LCOUNT,
     1   JLINE,ICO,IPRES1,IPRES2
      COMMON /BLCK5/ U44R,U44I,V44R,V44I,C44R,C44I,
     1   U4RNEW,U4INEW,V4RNEW,V4INEW,C4RNEW,C4INEW
      COMMON /BLCK6/ U55R,U55I,V55R,V55I,C55R,C55I,
     1   U5RNEW,U5INEW,V5RNEW,V5INEW,C5RNEW,C5INEW
C
      I = IHAVEP + 1
      XI = X(I)
      ARG = REDFRQ*W*XI
      U = COS(ARG)
      V = SIN(ARG)
C
C     IF POINT IS ON THE INITIAL RIGHT RUNNING MACH LINE,
C                   GO TO 2
C
      IF ( IHAVEP .NE. 0 ) GO TO 2
C
C     THIS PORTION IS FOR PITCH
C
      U2RNEW = -VRPANL*U - VIPANL*V
      U2INEW = -VIPANL*U + VRPANL*V
      V2RNEW = -U2RNEW
      V2INEW = -U2INEW
      C2RNEW = -U2RNEW
      C2INEW = -U2INEW
C
C     THIS PORTION IS FOR PLUNGE
```

```
C
      U4RNEW = -VRPLNG*U - VIPLNG*V
      U4INEW = -VIPLNG*U + VRPLNG*V
      V4RNEW = -U4RNEW
      V4INEW = -U4INEW
      C4RNEW = -U4RNEW
      C4INEW = -U4INEW
      RETURN
C
C     THIS PORTION IS FOR PITCH
C
    2 U22R(I) = VRPANL*U + VIPANL*V
      U22I(I) = VIPANL*U - VRPANL*V
      V22R(I) = U22R(I)
      V22I(I) = U22I(I)
      C22R(I) = -U22R(I)
      C22I(I) = -U22I(I)
C
C     THIS PORTION IS FOR PLUNGE
C
      U44R(I) = VRPLNG*U + VIPLNG*V
      U44I(I) = VIPLNG*U - VRPLNG*V
      V44R(I) = U44R(I)
      V44I(I) = U44I(I)
      C44R(I) = -U44R(I)
      C44I(I) = -U44I(I)
      RETURN
  100 FORMAT(2H M,7E16.7,/)
      END


C                     THIS IS SUBROUTINE TOP
C
      SUBROUTINE TOP
C
C     TOP COMPUTES U,V AND C AT AN UPPER SURFACE POINT
C
      REAL*8 X,Y,DSTSTR,HDSTRL,TRNGLH,STAG,FSTRMN
      REAL K12R,K12I,K34R,K34I,K56R,K56I
      DIMENSION X(400),Y(400)
      DIMENSION U22R(400),U22I(400),V22R(400),V22I(400),
     1   C22R(400),C22I(400)
      DIMENSION U44R(400),U44I(400),V44R(400),V44I(400),
     1   C44R(400),C44I(400)
      COMMON /BLCK1/ X,Y,REDFRQ,VRPANL,VIPANL,VRPLNG,VIPLNG,
     1   S,W,AI,BI
      COMMON /BLCK2/ U22R,U22I,V22R,V22I,C22R,C22I,
     1   U2RNEW,U2INEW,V2RNEW,V2INEW,C2RNEW,C2INEW
      COMMON /BLCK4/ IHAVEP,NGRDFN,NUM,NUMBLD,NSTPTS,LCOUNT,
     1   JLINE,ICO,IPRES1,IPRES2
      COMMON /BLCK5/ U44R,U44I,V44R,V44I,C44R,C44I,
     1   U4RNEW,U4INEW,V4RNEW,V4INEW,C4RNEW,C4INEW
      COMMON /GEOM/ FSTRMN,DSTSTR,HDSTRL,TRNGLH,STAG,XSUBO,
     1   DELTA,FAZE
C
      XN = FLOAT(NUM)
      J = IHAVEP
      I = IHAVEP + 1
      XNEW = SNGL(X(I)) - SNGL(STAG)*XN
      ARG = DELTA*XN
C     THIS PORTION IS FOR PITCH
C
C     DEFINE CONSTANTS. (J) SUBSCRIPT REFERS TO PERTURBATION
C     QUANTITIES ON THE PREVIOUS MACH LINE AT GRID POINT
C     IHAVEP. (NEW) INDICATES QUANTITIES ON THE CURRENT MACH
C     LINE AT IHAVEP
C
      K12R = U22R(J) + C22R(J) + AI*U22I(J)
      K12I = U22I(J) + C22I(J) - AI*U22R(J)
      K34R = -( COS(ARG) - ( XNEW - XSUBO )*REDFRQ*
```

```
     1               SIN(ARG) )/S
      K34I = -( SIN(ARG) + ( XNEW - XSUBO )*REDFRQ*
     1               COS(ARG) )/S
      K56R = U2RNEW + V2RNEW + BI*( U2INEW - C2INEW )
      K56I = U2INEW + V2INEW - BI*( U2RNEW - C2RNEW )
C
      G1 = 1. - AI*BI
      G2 = BI + BI
      G3 = G1*G1 + G2*G2
      G4 = K56R - K34R - BI*K12I
      G5 = K56I - K34I + BI*K12R
C
C     NOW STORE THE PERTURBATION QUANTITIES  FOR GRID POINT
C     IHAVEP ON THIS MACH LINE
C
      U22R(J)  = U2RNEW
      U22I(J)  = U2INEW
      V22R(J)  = V2RNEW
      V22I(J)  = V2INEW
      C22R(J)  = C2RNEW
      C22I(J)  = C2INEW
C
C     CCMPUTE PERTURBATION QUANTITIES AT THE CURRENT GRID
C     POINT (IHAVEP+1), AND TEMPORARILY STORE AS (NEW)
C
      U2RNEW  = ( G4*G1  + G5*G2 )/G3
      U2INEW  = (-G4*G2  + G5*G1 )/G3
      V2RNEW  = K34R
      V2INEW  = K34I
      C2RNEW  = K12R - U2RNEW + AI*U2INEW
      C2INEW  = K12I - U2INEW - AI*U2RNEW
C
C     THIS PORTION IS FOR PLUNGE. STEPS ARE THE SAME AS FOR
C     PITCH
C
      K12R = U44R(J) + C44R(J) + AI*U44I(J)
      K12I = U44I(J) + C44I(J) - AI*U44R(J)
      K34R = REDFRQ*SIN(ARG)/S
      K34I = -REDFRQ*COS(ARG)/S
      K56R = U4RNEW + V4RNEW + BI*( U4INEW - C4INEW )
      K56I = U4INEW + V4INEW - BI*( U4RNEW - C4RNEW )
      G4 = K56R - K34R - BI*K12I
      G5 = K56I - K34I + BI*K12R
C
      U44R(J)  = U4RNEW
      U44I(J)  = U4INEW
      V44R(J)  = V4RNEW
      V44I(J)  = V4INEW
      C44R(J)  = C4RNEW
      C44I(J)  = C4INEW
C
      U4RNEW  = ( G4*G1  + G5*G2 )/G3
      U4INEW  = (-G4*G2  + G5*G1 )/G3
      V4RNEW  = K34R
      V4INEW  = K34I
      C4RNEW  = K12R - U4RNEW + AI*U4INEW
      C4INEW  = K12I - U4INEW - AI*U4RNEW
C
      N = NUM + 1
      RETURN
  100 FORMAT(2H T,I2,7E16.7,/)
      END
C
C
C




C                THIS IS SUBROUTINE BOTTOM
C
      SUBROUTINE BOTTOM
```

```
      REAL*8 X,Y
      REAL K12R,K12I,K34R,K34I,K56R,K56I
      DIMENSION X(400),Y(400)
      DIMENSION U22R(400),U22I(400),V22R(400),V22I(400),
     1   C22R(400),C22I(400)
      DIMENSION U33R(400),U33I(400),V33R(400),V33I(400),
     1   C33R(400),C33I(400)
      DIMENSION U44R(400),U44I(400),V44R(400),V44I(400),
     1   C44R(400),C44I(400)
      DIMENSION U55R(400),U55I(400),V55R(400),V55I(400),
     1   C55R(400),C55I(400)
      COMMON /BLCK1/ X,Y,REDFRQ,VRPANL,VIPANL,VPPLNG,VIPLNG,
     1   S,W,AI,BI
      CCMMON /BLCK2/ U22R,U22I,V22R,V22I,C22R,C22I,
     1   U2RNEW,U2INEW,V2RNEW,V2INEW,C2RNEW,C2INEW
      COMMON /BLCK3/ U33R,U33I,V33R,V33I,C33R,C33I,
     1   U3RNEW,U3INEW,V3RNEW,V3INEW,C3RNEW,C3INEW
      COMMON /BLCK4/ IHAVEP,NGRDFN,NUM,NUMBLD,NSTPTS,LCOUNT,
     1   JLINE,ICO,IPRES1,IPRES2
      COMMON /BLCK5/ U44R,U44I,V44R,V44I,C44R,C44I,
     1   U4RNEW,U4INEW,V4RNEW,V4INEW,C4RNEW,C4INEW
      COMMON /BLCK6/ U55R,U55I,V55R,V55I,C55R,C55I,
     1   U5RNEW,U5INEW,V5RNEW,V5INEW,C5RNEW,C5INEW
C
C
C     BOTTOM COMPUTES U,V AND C AT A LOWER SURFACE POINT
C
      J = IHAVEP
      I = IHAVEP + 1
      XI = X(I)
C
C     THIS PORTION IS FOR PITCH
C
C     DEFINE CONSTANTS. (J) SUBSCRIPT REFERS TO PERTURBATION
C     QUANTITIES ON THE PREVIOUS MACH LINE AT GRID POINT
C     IHAVEP. (NEW) INDICATES QUANTITIES ON THE CURRENT MACH
C     LINE AT IHAVEP
C
      K12R = U33R(J) + C33R(J) + AI*U33I(J)
      K12I = U33I(J) + C33I(J) -AI*U33R(J)
      K34R = V2RNEW
      K34I = V2INEW
      K56R = U22R(I) - V22R(I) + BI*( U22I(I) - C22I(I) )
      K56I = U22I(I) - V22I(I) - BI*( U22R(I) - C22R(I) )
C
      G1 = 1. - AI*BI
      G2 = BI + BI
      G3 = G1*G1 + G2*G2
      G4 = K56R + K34R - BI*K12I
      G5 = K56I + K34I + BI*K12R
C
C     COMPUTE PERTURBATION QUANTITIES AT THE CURRENT GRID
C     POINT (IHAVEP+1), AND TEMPORARILY STORE AS (NEW)
C
      U3RNEW = ( G4*G1 + G5*G2 )/G3
      U3INEW = (-G4*G2 + G5*G1 )/G3
      V3RNEW = K34R
      V3INEW = K34I
      C3RNEW = K12R - U3RNEW + AI*U3INEW
      C3INEW = K12I - U3INEW - AI*U3RNEW
C
C     THIS PORTION IS FOR PLUNGE. STEPS ARE THE SAME AS FOR
C     PITCH
C
      K12R = U55R(J) + C55R(J) + AI*U55I(J)
      K12I = U55I(J) + C55I(J) -AI*U55R(J)
      K34R = V4RNEW
      K34I = V4INEW
      K56R = U44R(I) - V44R(I) + BI*( U44I(I) - C44I(I) )
      K56I = U44I(I) - V44I(I) - BI*( U44R(I) - C44R(I) )
C
      G4 = K56R + K34R - BI*K12I
```

```
      G5 = K56I + K34I + BI*K12R
C
      U5RNEW = ( G4*G1 + G5*G2 )/G3
      U5INEW = (-G4*G2 + G5*G1 )/G3
      V5RNEW = K34R
      V5INEW = K34I
      C5RNEW = K12R - U5RNEW + AI*U5INEW
      C5INEW = K12I - U5INEW - AI*U5RNEW
      RETURN
  100 FORMAT(2H B,7E16.7,/)
      END


C                THIS IS SUBROUTINE GENFPT
C
      SUBROUTINE GENFPT
C
C     GENFPT COMPUTES U,V, AND C AT A GENERAL FIELD POINT
C
      REAL*8 X,Y
      REAL K12R,K12I,K34R,K34I,K56R,K56I
      DIMENSION X(400),Y(400)
      DIMENSION U22R(400),U22I(400),V22R(400),V22I(400),
     1   C22R(400),C22I(400)
      DIMENSION U33R(400),U33I(400),V33R(400),V33I(400),
     1   C33R(400),C33I(400)
      DIMENSION U44R(400),U44I(400),V44R(400),V44I(400),
     1   C44R(400),C44I(400)
      DIMENSION U55R(400),U55I(400),V55R(400),V55I(400),
     1   C55R(400),C55I(400)
      COMMON /BLCK1/ X,Y,REDFRQ,VRPANL,VIPANL,VRPLNG,VIPLNG,
     1   S,W,AI,BI
      COMMON /BLCK2/ U22R,U22I,V22R,V22I,C22R,C22I,
     1   U2RNEW,U2INEW,V2RNEW,V2INEW,C2RNEW,C2INEW
      COMMON /BLCK3/ U33R,U33I,V33R,V33I,C33R,C33I,
     1   U3RNEW,U3INEW,V3RNEW,V3INEW,C3RNEW,C3INEW
      COMMON /BLCK4/ IHAVEP,NGRDFN,NUM,NUMBLD,NSTPTS,LCOUNT,
     1   JLINE,ICO,IPRES1,IPRES2
      COMMON /BLCK5/ U44R,U44I,V44R,V44I,C44R,C44I,
     1   U4RNEW,U4INEW,V4RNEW,V4INEW,C4RNEW,C4INEW
      COMMON /BLCK6/ U55R,U55I,V55R,V55I,C55R,C55I,
     1   U5RNEW,U5INEW,V5RNEW,V5INEW,C5RNEW,C5INEW
C
      J = IHAVEP
      I = IHAVEP + 1
      XI = X(I)
C
C     ICO  INSURES PROPER SELECTION OF FLOW FIELD
C     QUANTITIES AT F21 IF THE GRID POINT IS THE FIRST ONE
C     AFTER A BLADE OR WAKE ( ICO = 1 )
C
      IF ( ICO .NE. 1 ) GO TO 1
C
C     (NEW) REFERS TO THE PREVIOUS GRID POINT ON THIS MACH
C     LINE (IHAVEP). THIS GRID POINT IS ON THE LOWER SURFACE
C     OF A BLADE OR WAKE
C
      A = U3RNEW
      B = U3INEW
      C = C3RNEW
      D = C3INEW
      GO TO 2
C
    1 A = U2RNEW
      B = U2INEW
      C = C2RNEW
      D = C2INEW
C
C     THIS PORTION IS FOR PITCH
C
```

```
C     DEFINE CONSTANTS. (J) SUBSCRIPT REFERS TO PERTURBATION
C     QUANTITIES ON THE PREVIOUS MACH LINE AT GRID POINT
C     IHAVEP, (I) REFERS TO QUANTITIES ON THE PREVIOUS MACH
C     LINE AT (IHAVEP+1), WHILE (NEW) REFERS TO QUANTITIES
C     ON THE CURRENT MACH LINE AT (IHAVEP)
C
    2 K12R = U22R(J) + C22R(J) + AI*U22I(J)
      K12I = -AI*U22R(J) + U22I(J) + C22I(J)
      K34R = U22R(I) - V22R(I) + BI*( U22I(I) - C22I(I) )
      K34I = U22I(I) - V22I(I) - BI*( U22R(I) - C22R(I) )
      K56R = A + V2RNEW + BI*( B - D )
      K56I = B + V2INEW - BI*( A - C )
C
      G1 = .5*( K34R + K56R )
      G2 = BI*K12I
      G3 = 1. - AI*BI
      G4 = .5*( K34I + K56I )
      G5 = BI*K12R
      G6 = BI + BI
      G7 = G3*G3 + G6*G6
      G8 = G1 - G2
      G9 = G4 + G5
C
C     NOW STORE THE PERTURBATION QUANTITIES  FOR GRID POINT
C     IHAVEP ON THIS MACH LINE
C
      U22R(J) = U2RNEW
      U22I(J) = U2INEW
      V22R(J) = V2RNEW
      V22I(J) = V2INEW
      C22R(J) = C2RNEW
      C22I(J) = C2INEW
C
      IF ( ICO .EQ. 0 ) GO TO 3
      U33R(J) = U3RNEW
      U33I(J) = U3INEW
      V33R(J) = V3RNEW
      V33I(J) = V3INEW
      C33R(J) = C3RNEW
      C33I(J) = C3INEW
C
C     COMPUTE PERTURBATION QUANTITIES AT THE CURRENT GRID
C     POINT (IHAVEP+1), AND TEMPORARILY STORE AS (NEW)
C
    3 U2RNEW = ( G8*G3 + G9*G6 )/G7
      U2INEW = (-G8*G6 + G9*G3 )/G7
      V2RNEW = .5*( K56R - K34R )
      V2INEW = .5*( K56I - K34I )
      C2RNEW = K12R - U2RNEW + U2INEW*AI
      C2INEW = K12I - U2INEW - U2RNEW*AI
C
C     THIS PORTION IS FOR PLUNGE. STEPS ARE THE SAME AS FOR
C     PITCH
C
      IF ( ICO .NE. 1 ) GO TO 4
      A = U5RNEW
      B = U5INEW
      C = C5RNEW
      D = C5INEW
      GO TO 5
C
    4 A = U4RNEW
      B = U4INEW
      C = C4RNEW
      D = C4INEW
C
    5 K12R = U44R(J) + C44R(J) + AI*U44I(J)
      K12I = -AI*U44R(J) + U44I(J) + C44I(J)
      K34R = U44R(I) - V44R(I) + BI*( U44I(I) - C44I(I) )
      K34I = U44I(I) - V44I(I) - BI*( U44R(I) - C44R(I) )
      K56R = A + V4RNEW + BI*( B - D )
      K56I = B + V4INEW - BI*( A - C )
```

```
      G1 = .5*( K34R + K56R )
      G2 = BI*K12I
      G4 = .5*( K34I + K56I )
      G5 = BI*K12R
      G8 = G1 - G2
      G9 = G4 + G5
C
      U44R(J) = U4RNEW
      U44I(J) = U4INEW
      V44R(J) = V4RNEW
      V44I(J) = V4INEW
      C44R(J) = C4RNEW
      C44I(J) = C4INEW
C
      IF ( ICO .EQ. 0 ) GO TO 6
      U55R(J) = U5RNEW
      U55I(J) = U5INEW
      V55R(J) = V5RNEW
      V55I(J) = V5INEW
      C55R(J) = C5RNEW
      C55I(J) = C5INEW
C
C     RESET THE LOGIC VARIABLE TO INDICATE A GENERAL FIELD
C     POINT
C
      ICO = 0
C
    6 U4RNEW = ( G8*G3 + G9*G6 )/G7
      U4INEW = (-G8*G6 + G9*G3 )/G7
      V4RNEW = .5*( K56R - K34R )
      V4INEW = .5*( K56I - K34I )
      C4RNEW = K12R - U4RNEW + U4INEW*AI
      C4INEW = K12I - U4INEW - U4RNEW*AI
C
      RETURN
  100 FORMAT(2H G,7E16.7,/)
      END


C                    THIS IS SUBROUTINE NEWBLD
C
      SUBROUTINE NEWBLD
C
C     NEWBLD COMPUTES U,V AND C AT THE FIRST POSITION ON A B
C     ENCOUNTERED FOR THE FIRST TIME
C
      REAL*8 X,Y,DSTSTR,HDSTRL,TRNGLH,STAG,FSTRMN
      REAL K12R,K12I,K34R,K34I,K56R,K56I
      DIMENSION X(400),Y(400)
      DIMENSION U22R(400),U22I(400),V22R(400),V22I(400),
     1   C22R(400),C22I(400)
      DIMENSION U33R(400),U33I(400),V33R(400),V33I(400),
     1   C33R(400),C33I(400)
      DIMENSION U44R(400),U44I(400),V44R(400),V44I(400),
     1   C44R(400),C44I(400)
      DIMENSION U55R(400),U55I(400),V55R(400),V55I(400),
     1   C55R(400),C55I(400)
      COMMON /BLCK1/ X,Y,REDFRQ,VRPANL,VIPANL,VRPLNG,VIPLNG,
     1   S,W,AI,BI
      COMMON /BLCK2/ U22R,U22I,V22R,V22I,C22R,C22I,
     1   U2RNEW,U2INEW,V2RNEW,V2INEW,C2RNEW,C2INEW
      COMMON /BLCK3/ U33R,U33I,V33R,V33I,C33R,C33I,
     1   U3RNEW,U3INEW,V3RNEW,V3INEW,C3RNEW,C3INEW
      COMMON /BLCK4/ IHAVEP,NGRDFN,NUM,NUMBLD,NSTPTS,LCOUNT,
     1   JLINE,ICO,IPRES1,IPRES2
      COMMON /BLCK5/ U44R,U44I,V44R,V44I,C44R,C44I,
     1   U4RNEW,U4INEW,V4RNEW,V4INEW,C4RNEW,C4INEW
      COMMON /BLCK6/ U55R,U55I,V55R,V55I,C55R,C55I,
     1   U5RNEW,U5INEW,V5RNEW,V5INEW,C5RNEW,C5INEW
      COMMON /GEOM/ FSTRMN,DSTSTR,HDSTRL,TRNGLH,STAG,XSUBO,
```

```
     1   DELTA,FAZE
C
      J = IHAVEP
      I = IHAVEP + 1
      XI = X(I)
      NUM = NUM + 1
      XN = FLOAT(NUM)
C
C     THIS PORTION IS FOR PITCH
C
C     DEFINE CONSTANTS. (J) SUBSCRIPT REFERS TO PERTURBATION
C     QUANTITIES ON THE PREVIOUS MACH LINE AT GRID POINT
C     IHAVEP. (NEW) INDICATES QUANTITIES ON THE CURRENT MACH
C     LINE AT IHAVEP
      K12R = U22R(J) + C22R(J) + AI*U22I(J)
      K12I = -AI*U22R(J) + U22I(J) + C22I(J)
      K56R = U2RNEW + V2RNEW + BI*( U2INEW - C2INEW )
      K56I = U2INEW + V2INEW - BI*( U2RNEW - C2RNEW )
C
C     NOW STORE THE PERTURBATION QUANTITIES  FOR GRID POINT
C     IHAVEP ON THIS MACH LINE
C
      U22R(J) = U2RNEW
      U22I(J) = U2INEW
      V22R(J) = V2RNEW
      V22I(J) = V2INEW
      C22R(J) = C2RNEW
      C22I(J) = C2INEW
C
      ARG = DELTA*XN
C
C     THE FLOW TANGENCY CONDITION
C
      K34R = -( COS(ARG) + XSUBO*REDFRQ*SIN(ARG) )/S
      K34I = -( SIN(ARG) - XSUBO*REDFRQ*COS(ARG) )/S
C
      G1 = 1. - AI*BI
      G2 = BI + BI
      G3 = G1*G1 + G2*G2
      G4 = K56R - K34R - BI*K12I
      G5 = K56I - K34I + BI*K12R
C
C     COMPUTE PERTURBATION QUANTITIES AT THE CURRENT GRID
C     POINT (IHAVEP+1), AND TEMPORARILY STORE AS (NEW)
C
      U2RNEW = ( G4*G1 + G5*G2 )/G3
      U2INEW = (-G4*G2 + G5*G1 )/G3
      V2RNEW = K34R
      V2INEW = K34I
      C2RNEW = K12R - U2RNEW + AI*U2INEW
      C2INEW = K12I - U2INEW - AI*U2RNEW
C
      K34R = U22R(I) - V22R(I) + BI*( U22I(I) - C22I(I) )
      K34I = U22I(I) - V22I(I) - BI*( U22R(I) - C22R(I) )
      K56R = V2RNEW
      K56I = V2INEW
      G4 = K56R + K34R - BI*K12I
      G5 = K56I + K34I + BI*K12R
      U3RNEW = ( G4*G1 + G5*G2 )/G3
      U3INEW = (-G4*G2 + G5*G1 )/G3
      V3RNEW = K56R
      V3INEW = K56I
      C3RNEW = K12R - U3RNEW + AI*U3INEW
      C3INEW = K12I - U3INEW - AI*U3RNEW
C
C     THIS PORTION IS FOR PLUNGE. STEPS ARE THE SAME AS FOR
C     PITCH
C
      K12R = U44R(J) + C44R(J) + AI*U44I(J)
      K12I = -AI*U44R(J) + U44I(J) + C44I(J)
      K56R = U4RNEW + V4RNEW + BI*( U4INEW - C4INEW )
      K56I = U4INEW + V4INEW - BI*( U4RNEW - C4RNEW )
```

```
      U44R(J) = U4RNEW
      U44I(J) = U4INEW
      V44R(J) = V4RNEW
      V44I(J) = V4INEW
      C44R(J) = C4RNEW
      C44I(J) = C4INEW
C
      K34R = REDFRQ*SIN(ARG)/S
      K34I = - REDFRQ*COS(ARG)/S
      G4 = K56R - K34R - BI*K12I
      G5 = K56I - K34I + BI*K12R
C
      U4RNEW = ( G4*G1 + G5*G2 )/G3
      U4INEW = (-G4*G2 + G5*G1 )/G3
      V4RNEW = K34R
      V4INEW = K34I
      C4RNEW = K12R - U4RNEW + AI*U4INEW
      C4INEW = K12I - U4INEW - AI*U4RNEW
C
      K34R = U44R(I) - V44R(I) + BI*( U44I(I) - C44I(I) )
      K34I = U44I(I) - V44I(I) - BI*( U44R(I) - C44R(I) )
      K56R = V4RNEW
      K56I = V4INEW
      G4 = K56R + K34R - BI*K12I
      G5 = K56I + K34I + BI*K12R
      U5RNEW = ( G4*G1 + G5*G2 )/G3
      U5INEW = (-G4*G2 + G5*G1 )/G3
      V5RNEW = K56R
      V5INEW = K56I
      C5RNEW = K12R - U5RNEW + AI*U5INEW
      C5INEW = K12I - U5INEW - AI*U5RNEW
      NNBLD = NUM + 1
      RETURN
  100 FORMAT(2H N,7E16.7,/)
  101 FORMAT('0',10X,8H BLADE #,I3)
      END


C                    THIS IS SUBROUTINE WAKE
C
      SUBROUTINE WAKE
C
C     WAKE COMPUTES U,V AND C ABOVE AND BELOW THE SLIP PLANE
C     AFT OF A BLADE.
C
      REAL*8 X,Y
      DIMENSION X(400),Y(400)
      DIMENSION U22R(400),U22I(400),V22R(400),V22I(400),
     1   C22R(400),C22I(400)
      DIMENSION U33R(400),U33I(400),V33R(400),V33I(400),
     1   C33R(400),C33I(400)
      DIMENSION U44R(400),U44I(400),V44R(400),V44I(400),
     1   C44R(400),C44I(400)
      DIMENSION U55R(400),U55I(400),V55R(400),V55I(400),
     1   C55R(400),C55I(400)
      DIMENSION A(8,8),B(8)
      COMMON /BLCK1/ X,Y,REDFRQ,VRPANL,VIPANL,VRPLNG,VIPLNG,
     1   S,W,AI,BI
      COMMON /BLCK2/ U22R,U22I,V22R,V22I,C22R,C22I,
     1   U2RNEW,U2INEW,V2RNEW,V2INEW,C2RNEW,C2INEW
      COMMON /BLCK3/ U33R,U33I,V33R,V33I,C33R,C33I,
     1   U3RNEW,U3INEW,V3RNEW,V3INEW,C3RNEW,C3INEW
      COMMON /BLCK4/ IHAVEP,NGRDFN,NUM,NUMBLD,NSTPTS,LCOUNT,
     1   JLINE,ICO,IPRES1,IPRES2
      COMMON /BLCK5/ U44R,U44I,V44R,V44I,C44R,C44I,
     1   U4RNEW,U4INEW,V4RNEW,V4INEW,C4RNEW,C4INEW
      COMMON /BLCK6/ U55R,U55I,V55R,V55I,C55R,C55I,
     1   U5RNEW,U5INEW,V5RNEW,V5INEW,C5RNEW,C5INEW
C
      J = IHAVEP
```

```
      I = IHAVEP + 1
      XI = X(I)
C
C     ITIME TELLS WHETHER PLUNGE SOLUTION HAS BEEN DONE. IF
C     ITIME = 2, PLUNGE SOLUTION IS COMPLETE; IF ITIME = 1,
C     PLUNGE SOLUTION NEEDS TO BE DONE
C
      ITIME = 1
C
C     THIS PORTION IS FOR PITCH
C
C     COMPUTE THE RIGHT HAND SIDE OF THE EQUATIONS. (I) AND
C     (J) SUBSCRIPTS REFER TO PERTURBATION QUANTITIES AT
C     GRID POINTS (IHAVEP+1) AND (IHAVEP) ON THE PREVIOUS
C     MACH LINE, WHILE (NEW) REFERS TO GRID POINT (IHAVEP)
C     ON THE CURRENT MACH LINE
C
      B(1) = U22R(J) + C22R(J) + AI*U22I(J)
      B(2) = U22I(J) + C22I(J) - AI*U22R(J)
      B(3) = U2RNEW + V2RNEW + BI*( U2INEW - C2INEW )
      B(4) = U2INEW + V2INEW - BI*( U2RNEW - C2RNEW )
      B(5) = U33R(J) + C33R(J) + AI*U33I(J)
      B(6) = U33I(J) + C33I(J) -AI*U33R(J)
      B(7) = U22R(I) - V22R(I) + BI*( U22I(I) - C22I(I) )
      B(8) = U22I(I) - V22I(I) - BI*( U22R(I) - C22R(I) )
C
C     NOW STORE THE PERTURBATION QUANTITIES  FOR GRID POINT
C     IHAVEP ON THIS MACH LINE
C
      U22R(J) = U2RNEW
      U22I(J) = U2INEW
      V22R(J) = V2RNEW
      V22I(J) = V2INEW
      C22R(J) = C2RNEW
      C22I(J) = C2INEW
C
C     SET UP THE MATRIX OF COEFFICIENTS
C
    1 DO 2 K = 1,8
      DO 2 L = 1,8
    2 A(K,L) = 0.
      A(1,1) = 1.
      A(1,7) = 1.
      A(2,2) = 1.
      A(2,8) = 1.
      A(3,1) = 1.
      A(3,5) = 1.
      A(4,2) = 1.
      A(4,6) = 1.
      A(5,3) = 1.
      A(5,7) = 1.
      A(6,4) = 1.
      A(6,8) = 1.
      A(7,3) = 1.
      A(8,4) = 1.
      A(1,2) = -AI
      A(5,4) = -AI
      A(6,3) = AI
      A(2,1) = AI
      A(3,2) = -BI
      A(4,7) = -BI
      A(7,4) = -BI
      A(8,7) = -BI
      A(3,8) = BI
      A(4,1) = BI
      A(7,8) = BI
      A(8,3) = BI
      A(7,5) = -1.
      A(8,6) = -1.
C
C     CALL SIMULTANEOUS LINEAR EQUATION SOLVER TO CALCULATE
```

```
C     U, V AND C  AT THIS GRID POINT (IHAVEP+1)
C
      CALL SIMQ(A,B,8,KS)
C
      IF ( ITIME .EQ. 2 ) GO TO 3
      ITIME = 2
C
      U2RNEW = B(1)
      U2INEW = B(2)
      U3RNEW = B(3)
      U3INEW = B(4)
      V2RNEW = B(5)
      V2INEW = B(6)
      C2RNEW = B(7)
      C2INEW = B(8)
      V3RNEW = B(5)
      V3INEW = B(6)
      C3RNEW = B(7)
      C3INEW = B(8)
C     COMPUTE THE RIGHT HAND SIDE OF THE EQUATIONS
C
C
C     THIS PORTION IS FOR PLUNGE. STEPS ARE THE SAME AS FOR
C     PITCH
C
      B(1) = U44R(J) + C44R(J) + AI*U44I(J)
      B(2) = U44I(J) + C44I(J) - AI*U44R(J)
      B(3) = U4RNEW + V4RNEW + BI*( U4INEW - C4INEW )
      B(4) = U4INEW + V4INEW - BI*( U4RNEW - C4RNEW )
      B(5) = U55R(J) + C55R(J) + AI*U55I(J)
      B(6) = U55I(J) + C55I(J) -AI*U55R(J)
      B(7) = U44R(I) - V44R(I) + BI*( U44I(I) - C44I(I) )
      B(8) = U44I(I) - V44I(I) - BI*( U44R(I) - C44R(I) )
C
      U44R(J) = U4RNEW
      U44I(J) = U4INEW
      V44R(J) = V4RNEW
      V44I(J) = V4INEW
      C44R(J) = C4RNEW
      C44I(J) = C4INEW
C
C     SET UP THE MATRIX OF COEFFICIENTS
C
      GO TO 1
C
    3 U4RNEW = B(1)
      U4INEW = B(2)
      U5RNEW = B(3)
      U5INEW = B(4)
      V4RNEW = B(5)
      V4INEW = B(6)
      C4RNEW = B(7)
      C4INEW = B(8)
      V5RNEW = B(5)
      V5INEW = B(6)
      C5RNEW = B(7)
      C5INEW = B(8)
C
      ITIME = 1
C
      RETURN
  100 FORMAT(2H W,7E16.7,/)
      END


C                THIS IS SUBROUTINE SIMQ
C
C     THIS IS SUBROUTINE SIMQ
      SUBROUTINE SIMQ(A,B,N,KS)
C
```

```
C     SIMQ IS A SIMULTANEOUS LINEAR EQUATION SOLVER
C
C         A   -  MATRIX (N BY N) OF COEFFICIENTS STORED
C                COLUMNWISE (DESTROYED)
C         B   -  VECTOR OF ORIGINAL CONSTANTS.  FINAL SOLUTION
C                ON THIS VECTOR.
C         N   -  NUMBER OF EQUATIONS AND VARIABLES
C                MUST BE > 1
C         KS  -  OUTPUT DIGIT
C                0  FOR NORMAL SOLUTION
C                1  FOR A SINGULAR SOLUTION
C
      DIMENSION A(1),B(1)
C
C     FORWARD SOLUTION
C
      TOL = 0.
      KS = 0
      JJ = -N
      DO 65 J = 1,N
      JY = J + 1
      JJ = JJ + N + 1
      BIGA = 0.
      IT = JJ - J
      DO 30 I = J,N
C
C     SEARDH FOR MAXIMUM COEFFICIENT IN COLUMN
C
      IJ = IT + I
      IF (ABS(BIGA) - ABS(A(IJ)) ) 20,30,30
   20 BIGA = A(IJ)
      IMAX = I
   30 CONTINUE
C
C     TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)
C
      IF (ABS(BIGA) - TOL) 35,35,40
   35 KS = 1
      RETURN
C
C     INTERCHANGE ROWS IF NECESSARY
C
   40 I1 = J + N*(J-2)
      IT = IMAX - J
      DO 50 K = J,N
      I1 = I1 + N
      I2 = I1 + IT
      SAVE = A(I1)
      A(I1) = A(I2)
      A(I2) = SAVE
C
C     DIVIDE EQUATION BY LEADING COEFFICIENT
C
   50 A(I1) = A(I1)/BIGA
      SAVE = B(IMAX)
      B(IMAX) = B(J)
      B(J) = SAVE/BIGA
C
C     ELIMINATE THE NEXT VARIABLE
C
      IF (J-N) 55,70,55
   55 IQS = N*(J-1)
      DO 65 IX = JY,N
      IXJ = IQS + IX
      IT = J - IX
      DO 60 JX = JY,N
      IXJX = N*(JX-1) + IX
      JJX = IXJX + IT
   60 A(IXJX) = A(IXJX) - (A(IXJ)*A(JJX))
   65 B(IX) = B(IX) - (B(J)*A(IXJ))
C
C     BACK SOLUTION
```

```
C
   70 NY = N-1
      IT = N*N
      DO 80 J = 1,NY
      IA = IT - J
      IB = N - J
      IC = N
      DO 80 K = 1,J
      B(IB) = B(IB) - A(IA)*B(IC)
      IA = IA - N
   80 IC = IC - 1
C
      RETURN
      END


C                   THIS IS SUBROUTINE COEF
C
      SUBROUTINE COEF
C
C     SUBROUTINE COEF COMPUTES THE NON-DIMENSIONAL LIFT
C     FORCE AND MOMENT FOR EACH BLADE. IN ADDITION, THE
C     MAGNITUDE AND PHASE OF THE COMPLEX LIFT FORCE AND
C     MOMENT ARE DETERMINED
C
      REAL*8 X,Y,FSTRMN,DSTSTR,HDSTRL,TRNGLH,STAG,XNEW,XOLD,
     1   XN
      DIMENSION XPT(4,100),PR12(4,100),PI12(4,100),RMR12(4,
     1   100),RMI12(4,100),PR34(4,100),PI34(4,100),RMR34(4,
     2100),RMI34(4,100)
      DIMENSION RL1(76),RL2(76),RL3(76),RL4(76),RM1(76),
     1   RM2(76),RM3(76),RM4(76),ANGP12(76),ANGM12(76),
     2   ANGP34(76),ANGM34(76),P12MAG(76),P34MAG(76),
     3   R12MAG(76),R34MAG(76)
      DIMENSION X(400),Y(400)
      DIMENSION U22R(400),U22I(400),V22R(400),V22I(400),
     1   C22R(400),C22I(400)
      DIMENSION U33R(400),U33I(400),V33R(400),V33I(400),
     1   C33R(400),C33I(400)
      DIMENSION U44R(400),U44I(400),V44R(400),V44I(400),
     1   C44R(400),C44I(400)
      DIMENSION U55R(400),U55I(400),V55R(400),V55I(400),
     1   C55R(400),C55I(400)
      DIMENSION NN(100),ID(76),LB(76)
      COMMON /BLCK1/ X,Y,REDFRQ,VRPANL,VIPANL,VRPLNG,VIPLNG,
     1   S,W,AI,BI
      COMMON /BLCK2/ U22R,U22I,V22R,V22I,C22R,C22I,
     1   U2RNEW,U2INEW,V2RNEW,V2INEW,C2RNEW,C2INEW
      COMMON /BLCK3/ U33R,U33I,V33R,V33I,C33R,C33I,
     1   U3RNEW,U3INEW,V3RNEW,V3INEW,C3RNEW,C3INEW
      COMMON /BLCK4/ IHAVEP,NGRDFN,NUM,NUMBLD,NSTPTS,LCOUNT,
     1   JLINE,ICO,IPRES1,IPRES2
      COMMON /BLCK5/ U44R,U44I,V44R,V44I,C44R,C44I,
     1   U4RNEW,U4INEW,V4RNEW,V4INEW,C4RNEW,C4INEW
      COMMON /BLCK6/ U55R,U55I,V55R,V55I,C55R,C55I,
     1   U5RNEW,U5INEW,V5RNEW,V5INEW,C5RNEW,C5INEW
      COMMON/BLCK7/ RL1,RL2,RL3,RL4,RM1,RM2,RM3,RM4,ANGP12,
     1   ANGM12,ANGP34,ANGM34,P12MAG,R12MAG,P34MAG,R34MAG
      COMMON/BLCK8/ XPT,PR12,PI12,PR34,PI34,IWRITE,LVEC,
     1   NITOT,NJUNK
      COMMON /GEOM/ FSTRMN,DSTSTR,HDSTRL,TRNGLH,STAG,XSUBO,
     1   DELTA,FAZE
      DATA LB/1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,
     1   4,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,
     2   1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,1,
     3   2,3,4/
C
C
      ENDPT(Y2,Y1,X0,X2,DELX) = ( Y2 - Y1 )*( X0 -
     1   X2 )/DELX + Y2
```

```
      XARM(V) = V - XSUBO
C
C
C     INITIALIZE THE ARRAYS AND SET VALUES AT THE LEADING
C     EDGE OF THE FIRST BLADE. CALLED AFTER  INITAL.  ID(L)
C     IS A BLADE DEPENDENT SWITCH VARIABLE:
C        = 0  GRID POINT IS ON THE BLADE -- MAKE CALCULATION
C        = 1  GRID POINT IS IN THE WAKE -- SKIP IT
C
      DO 10 IL = 1,4
      DO 5 IJ = 1,100
      NN(IJ) = 0
      XPT(IL,IJ) = 0.0
      PR12(IL,IJ) = 0.
      PI12(IL,IJ) = 0.
      PR34(IL,IJ) = 0.
      PI34(IL,IJ) = 0.
      RMR12(IL,IJ) = 0.
      RMI12(IL,IJ) = 0.
      RMR34(IL,IJ) = 0.
    5 RMI34(IL,IJ) = 0.
   10 CONTINUE
C
      DO 11 IL = 1,76
   11 ID(IL) = 0
C
      NN(1) = 1
      A1 = 0.5*( C55R(1) - C44R(1) )
      A2 = 0.5*( C55I(1) - C44I(1) )
      A3 = 0.5*( C33R(1) - C22R(1) )
      A4 = 0.5*( C33I(1) - C22I(1) )
      PR12(1,1) = A1
      PI12(1,1) = A2
      PR34(1,1) = A3
      PI34(1,1) = A4
      RMR12(1,1) = - XSUBO*A1
      RMI12(1,1) = - XSUBO*A2
      RMR34(1,1) = -XSUBO*A3
      RMI34(1,1) = -XSUBO*A4
      XPT(1,1) = SNGL( X(1) )
      RETURN
C
      ENTRY COEF1
C
C     THIS SECTION CALCULATES DELTA PRESSURE AND DELTA
C     MOMENT FOR EACH GRID POINT ON THE BLADE. CALLED AFTER
C     SUBROUTINES BOTTOM OR NEWBLD.
C
      M = NUM + 1
      I = IHAVEP + 1
      L = LB(M)
      NN(L) = NN(L) + 1
      NI = NN(L)
      XN = DFLOAT( NUM )
      XNEW = X(I) - STAG*XN
      XPT(L,NI) = SNGL( XNEW )
      A1 = 0.5*( C5RNEW - C4RNEW )
      A2 = 0.5*( C5INEW - C4INEW )
      A3 = 0.5*( C3RNEW - C2RNEW )
      A3 = 0.5*( C3INEW - C2INEW )
C
C     ARG1 IS USED TO COMPUTE THE PRESSURE ON THE N(TH)
C     BLADE WHEN IN THE INITIAL POSITION.
C
C     ARG2 IS USED TO COMPUTE THE PRESSURE ON THE N(TH)
C     BLADE WHEN IN THE MEAN POSITION, PITCHING LEADING
C     EDGE UP.
C
      ARG1 = - DELTA*SNGL(XN)
      ARG2 = 4.712389 - DELTA*SNGL(XN)
      CARG1 = COS(ARG1)
      CARG2 = COS(ARG2)
```

```
      SARG1 = SIN(ARG1)
      SARG2 = SIN(ARG2)
C
      PR12(L,NI) = A1*CARG1 - A2*SARG1
      PI12(L,NI) = A1*CARG2 - A2*SARG2
      PR34(L,NI) = A3*CARG1 - A4*SARG1
      PI34(L,NI) = A3*CARG2 - A4*SARG2
      RMR12(L,NI) = PR12(L,NI)*XARM(XNEW)
      RMI12(L,NI) = PI12(L,NI)*XARM(XNEW)
      RMR34(L,NI) = PR34(L,NI)*XARM(XNEW)
      RMI34(L,NI) = PI34(L,NI)*XARM(XNEW)
      GO TO 35
C
      ENTRY COEF2
C
C     FOR A PARTICULAR BLADE.  THE VALUES OF DELTA PRESSURE
C     AND MOMENT ARE EXTRAPOLATED LINEARLY. THIS SECTION IS
C     CALLED AND USED THE FIRST TIME  WAKE  IS USED FOR EACH
C     BLADE.
C
      M = NUM + 1
      IWRITE = 0
C
C     IF THIS IS NOT THE FIRST CALL FROM THE WAKE AFT OF
C     THIS BLADE, RETURN
C
      IF ( ID(M) .EQ. 1 )  GO TO 35
      I = IHAVEP + 1
      L = LB(M)
      NN(L) = NN(L) + 1
      NI = NN(L)
      NIM = NI - 1
      ID(M) = 1
C
C     THESE THREE VARIABLES ARE USED IN THE MAIN PROGRAM IF
C     THE PRESSURE DISTRIBUTION IS OUTPUT.
C        IWRITE = 1 -- CALCULATIONS FOR BLADE ARE COMPLETE
C        NITOT  --  TOTAL NUMBER OF GRID POINTS ON BLADE
C        LVEC   --  ENTRY POINT FOR THE STORAGE ARRAYS
C
      IWRITE = 1
      NITOT = NI
      LVEC = L
C
      XN = DFLOAT(NUM)
      XNEW = X(I) - STAG*XN
      XPT(L,NI) = 1.0
      XPT(L,1) = 0.
      XOLD = XNEW - DSTSTR
C
C     TRAPEZOIDAL INTEGRATION FOR LIFT FORCE AND MOMENT
C
      SUM1 = 0.
      SUM2 = 0.
      SUM3 = 0.
      SUM4 = 0.
      SUM5 = 0.
      SUM6 = 0.
      SUM7 = 0.
      SUM8 = 0.
      NI2 = NI - 2
      DO 71 IJ = 2,NI2
      SUM1 = SUM1 + PR12(L,IJ)
      SUM2 = SUM2 + PI12(L,IJ)
      SUM3 = SUM3 + RMR12(L,IJ)
      SUM4 = SUM4 + RMI12(L,IJ)
      SUM5 = SUM5 + PR34(L,IJ)
      SUM6 = SUM6 + PI34(L,IJ)
      SUM7 = SUM7 + RMR34(L,IJ)
   71 SUM8 = SUM8 + RMI34(L,IJ)
C
      SUM1 = ( SUM1 + 0.5*( PR12(L,1) + PR12(L,NIM) ) )*
```

```
     1             DSTSTR
      SUM2 = ( SUM2 + 0.5*( PI12(L,1) + PI12(L,NIM) ) )*
     1             DSTSTR
      SUM3 = ( SUM3 + 0.5*( RMR12(L,1) + RMR12(L,NIM) ) )*
     1             DSTSTR
      SUM4 = ( SUM4 + 0.5*( RMI12(L,1) + RMI12(L,NIM) ) )*
     1             DSTSTR
      SUM5 = ( SUM5 + 0.5*( PR34(L,1) + PR34(L,NIM) ) )*
     1             DSTSTR
      SUM6 = ( SUM6 + 0.5*( PI34(L,1) + PI34(L,NIM) ) )*
     1             DSTSTR
      SUM7 = ( SUM7 + 0.5*( RMR34(L,1) + RMR34(L,NIM) ) )*
     1             DSTSTR
      SUM8 = ( SUM8 + 0.5*( RMI34(L,1) + RMI34(L,NIM) ) )*
     1             DSTSTR
C
C     EXTRAPOLATE FOR VALUES AT THE END POINT OF THE BLADE.
C
      PR12(L,NI) = ENDPT(PR12(L,NIM),PR12(L,NI2),1.,XOLD,
     1            DSTSTR)
      PI12(L,NI) = ENDPT(PI12(L,NIM),PI12(L,NI2),1.,XOLD,
     1            DSTSTR)
      PR34(L,NI) = ENDPT(PR34(L,NIM),PR34(L,NI2),1.,XOLD,
     1            DSTSTR)
      PI34(L,NI) = ENDPT(PI34(L,NIM),PI34(L,NI2),1.,XOLD,
     1            DSTSTR)
C
      RMR12(L,NI) = PR12(L,NI)*(1. - XSUBO)
      RMI12(L,NI) = PI12(L,NI)*(1. - XSUBO)
      RMR34(L,NI) = PR34(L,NI)*(1. - XSUBO)
      RMI34(L,NI) = PI34(L,NI)*(1. - XSUBO)
      C = ( 1.0 - XOLD )*0.5
C
      RL1(M) = SUM1 + ( PR12(L,NI) + PR12(L,NIM) )*C
      RL2(M) = SUM2 + ( PI12(L,NI) + PI12(L,NIM) )*C
      RM1(M) = SUM3 + ( RMR12(L,NI) + RMR12(L,NIM) )*C
      RM2(M) = SUM4 + ( RMI12(L,NI) + RMI12(L,NIM) )*C
      RL3(M) = SUM5 + ( PR34(L,NI) + PR34(L,NIM) )*C
      RL4(M) = SUM6 + ( PI34(L,NI) + PI34(L,NIM) )*C
      RM3(M) = SUM7 + ( RMR34(L,NI) + RMR34(L,NIM) )*C
      RM4(M) = SUM8 + ( RMI34(L,NI) + RMI34(L,NIM) )*C
C
C     PHASE ANGLE
C
      ANGP12(M) = ( ATAN2( RL2(M),RL1(M) ) )*57.29578
      ANGM12(M) = ( ATAN2( RM2(M),RM1(M) ) )*57.29578
      ANGP34(M) = ( ATAN2( RL4(M),RL3(M) ) )*57.29578
      ANGM34(M) = ( ATAN2( RM4(M),RM3(M) ) )*57.29578
C
C     MAGNITUDE
C
      P12MAG(M) = SQRT( RL1(M)*RL1(M) + RL2(M)*RL2(M) )
      R12MAG(M) = SQRT( RM1(M)*RM1(M) + RM2(M)*RM2(M) )
      P34MAG(M) = SQRT( RL3(M)*RL3(M) + RL4(M)*RL4(M) )
      R34MAG(M) = SQRT( RM3(M)*RM3(M) + RM4(M)*RM4(M) )
C
      NN(L) = 0
C
   35 RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                          C
C         CYLINDRICAL SHELL/PANEL FLUTTER PROGRAM          C
C                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C                  THIS IS THE MAIN PROGRAM
C
      COMPLEX PHI(400),DPHIS1(400),DPHIS2(400)
      DIMENSION X(400),R(400),XPT(400),CPR(400),CPI(400)
      DIMENSION DUMMY(400),DUNCE(400)
      COMMON/BLCK1/ FSTRMN,DELTAS,DX,DH,DSTSTR,RO,RI,REDFRQ
      COMMON/BLCK2/ A1,A2,A3,A4,RN,RM,RM1,FACT1,FACT2
      COMMON/BLCK3/ NGRDFN,NPTS,LCOUNT,IHAVEP,N,M,M1,IPRINT
      COMMON/BLCK4/ PHI,DPHIS1,DPHIS2
      COMMON/BLCK5/ X,R,XPT,CPR,CPI
      DATA PI/3.141593/
C
      ENDPT(Y2,Y1,X0,X2,DELX) = ( Y2 - Y1 )*( X0 -X2 )/DELX
     1   + Y2
C
      EPS = 1.E-04
      IER = 0
      CALL INPUT(IER)
      IF ( IER .EQ. 1 ) GO TO 12
 1000 CONTINUE
C
      IHAVEP = 1
      LCOUNT = 1
C
      CALL INITAL
C
    1 CONTINUE
C
      CALL COMPXR
C
C
C     IS THE GRID POINT ON THE INITIAL RIGHT RUNNING MACH
C     LINE?
C
      IF (( LCOUNT .EQ. 1 ) .AND. (IHAVEP .LE. NGRDFN ))
     1   GO TO 3
C
C     IS THE GRID POINT ON THE OUTER CYLINDER SURFACE?
C
      IF ( IHAVEP .EQ. 0 ) GO TO 5
C
C     IS THE GRID POINT ON THE INNER CYLINDER SURFACE?
C
      IF ( IHAVEP .EQ. NGRDFN ) GO TO 6
C
C     THEN THE GRID POINT IS IN THE GENERAL FLOW FIELD.
C
    2 CALL GENFPT(IER)
      IF ( IER .EQ. 1 ) GO TO 12
      IHAVEP = IHAVEP + 1
      GO TO 1
C
    3 CALL MACHLN
      IF ( IHAVEP .EQ. NGRDFN ) GO TO 4
      IHAVEP = IHAVEP + 1
      GO TO 1
C
C     THIS IS THE LAST POINT ON THE INITIAL MACH LINE.
C
    4 LCOUNT = LCOUNT + 1
      IHAVEP = 0
      GO TO 1
C
    5 CALL RAD2
C
C     IS THIS GRID POINT PAST THE FLEXIBLE CYLINDER LENGTH?
C
```

```
      IF ( (X(1)+EPS) .GE. 1. ) GO TO 8
      IHAVEP = IHAVEP + 1
      GO TO 1
C
    6 CALL RAD1
      IHAVEP = 0
      LCOUNT = LCOUNT + 1
      GO TO 1
C
    8 CONTINUE
      IF ( ABS(XPT(LCOUNT) - 1. ) .LE. EPS ) GO TO 10
      L = LCOUNT
      LM1 = LCOUNT - 1
      LM2 = LCOUNT - 2
      XPT(L) = 1.
      CPR(L) = ENDPT(CPR(LM1),CPR(LM2),1.00,XPT(LM1),DSTSTR)
      CPI(L) = ENDPT(CPI(LM1),CPI(LM2),1.00,XPT(LM1),DSTSTR)
C
C
      WRITE(6,105)
      DO 9 I = 1,L
      ARG = RM1*PI*XPT(I)
      WRITE(6,110) I,XPT(I),CPR(I),CPI(I)
      DUMMY(I) = CPR(I)*SIN(ARG)
    9 DUNCE(I) = CPI(I)*SIN(ARG)
C
      SUM1 = 0.5*(CPR(L) + CPR(LM1))*(1.00 - XPT(LM1))
      SUM2 = 0.5*(CPI(L) + CPI(LM1))*(1.00 - XPT(LM1))
C
      WRITE(6,114) FSTRMN,REDFRQ,RO,RI,N
C
      CALL QSF(DSTSTR,DUMMY,DUMMY,LM1)
      CALL QSF(DSTSTR,DUNCE,DUNCE,LM1)
      QREAL = 0.5*(DUMMY(LM1) + SUM1)
      QIMAG = 0.5*(DUNCE(LM1) + SUM2)
C
      WRITE(6,115) M,M1,QREAL,M,M1,QIMAG
      GO TO 12
C
   10 CONTINUE
      XPT(L) = 1.
      WRITE(6,105)
      DO 11 I = 1,LCOUNT
      ARG = RM1*PI*XPT(I)
      WRITE(6,110) I,XPT(I),CPR(I),CPI(I)
      DUMMY(I) = CPR(I)*SIN(ARG)
   11 DUNCE(I) = CPI(I)*SIN(ARG)
C
      WRITE(6,114) FSTRMN,REDFRQ,RO,RI,N
C
      CALL QSF(DSTSTR,DUMMY,DUMMY,L)
      CALL QSF(DSTSTR,DUNCE,DUNCE,L)
      QREAL = 0.5*DUMMY(L)
      QIMAG= 0.5*DUNCE(L)
      WRITE(6,115) M,M1,QREAL,M,M1,QIMAG
C
   12 CONTINUE
      STOP
  150 FORMAT(F10.5)
  105 FORMAT('1',T19,'I',T27,'X',T38,'CPR',T50,'CPI',//)
  110 FORMAT('0',15X,I5,3(3X,F9.5))
  114 FORMAT(//' ',T20,'MACH NUMBER = ',F10.7,//' ',T20,
     1    'REDUCED FREQUENCY = ',F10.7,/' ',T20,'(RADIUS/',
     2    'LENGTH) OUTER = ',F10.7,/' ',T20,'(RADIUS/LENGTH)'
     3    ,' INNER = ',F10.7,/' ',T20,'CIRCUMFERENTIAL MODE',
     4    /' ',T34,'NUMBER = ',I5,//)
  115 FORMAT('0',T20,'Q(',I2,',',I2,')REAL = ',F10.5,/'0',
     1    T20,'Q(',I2,',',I2,')IMAG = ',F10.5,/)
  136 FORMAT('1')
      END
```

```
C                THIS IS SUBROUTINE INPUT
C
      SUBROUTINE INPUT(IER)
C
C     SUBROUTINE INPUT READS INPUT VARIABLES, AND DEFINES
C     SEVERAL CONSTANTS USED THROUGHOUT THE PROGRAM.
C
      COMMON/BLCK1/ FSTRMN,DELTAS,DX,DH,DSTSTR,RO,RI,REDFRQ
      COMMON/BLCK2/ A1,A2,A3,A4,RN,RM,RM1,FACT1,FACT2
      COMMON/BLCK3/ NGRDFN,NPTS,LCOUNT,IHAVEP,N,M,M1,IPRINT
      NAMELIST/NAM1/ FSTRMN,REDFRQ,RO,RI,NGRDFN,N,M,M1,
     1   IPRINT
C
      WRITE(6,80)
      READ(5,NAM1)
      WRITE(6,NAM1)
C
      FMANGL = ARSIN(1./FSTRMN)
      DH = (RO - RI)/FLOAT(NGRDFN)
C
      IF ( DH .LT. 0. ) GO TO 1
C
      DELTAS = DH*FSTRMN
      DX = DH/TAN(FMANGL)
      DSTSTR = 2.*DX
      NPTS = 1./DSTSTR + 1
C
      IF ( NPTS .GT. 400 ) GO TO 2
C
      BETA = SQRT(FSTRMN*FSTRMN - 1.)
      RN = FLOAT(N)
      RM = FLOAT(M)
      RM1 = FLOAT(M1)
      FACT1 = REDFRQ*REDFRQ
      FACT2 = RN*RN/(FSTRMN*FSTRMN)
C
C     COMPUTE CONSTANTS TO BE USED IN THE COMPUTATIONAL
C     MOLECULES.
C
      A1 = 0.25*DELTAS/FSTRMN
      A2 = 0.5*FSTRMN/BETA
      A3 = REDFRQ*DELTAS*A2
      A4 = 0.5*FSTRMN
      RETURN
C
    1 IER = 1
      WRITE(6,100)
      GO TO 3
C
    2 IER = 1
      WRITE(6,110)
    3 RETURN
   80 FORMAT('1')
  100 FORMAT(//' ',35X,'ABNORMAL TERMINATION: INPUT RADII ',
     1   'INCORRECT',//)
  110 FORMAT(//' ',35X,'ABNORMAL TERMINATION: MUST INCREAS',
     1   'E THE SIZE OF VECTORS XPT, CPR, AND CPI',//)
      END


C                THIS IS SUBROUTINE INITAL
C
      SUBROUTINE INITAL
C
C     SUBROUTINE INITAL INITIALIZES ALL FLOW FIELD
C     QUANTITIES.
C
      COMPLEX DPHIDR,DPHIDX
      COMPLEX PHI(400),DPHIS1(400),DPHIS2(400)
      DIMENSION X(400),R(400),XPT(400),CPR(400),CPI(400)
```

```
      COMMON/BLCK1/ FSTRMN,DELTAS,DX,DH,DSTSTR,RO,RI,REDFRQ
      COMMON/BLCK2/ A1,A2,A3,A4,RN,RM,RM1,FACT1,FACT2
      COMMON/BLCK3/ NGRDFN,NPTS,LCOUNT,IHAVEP,N,M,M1,IPRINT
      COMMON/BLCK4/ PHI,DPHIS1,DPHIS2
      COMMON/BLCK5/ X,R,XPT,CPR,CPI
      DATA PI/3.141593/
C
      X(1) = 0.
      R(1) = RO
      XPT(1) = 0.
C
C     ALONG THE INITIAL RIGHT RUNNING CHARACTERISTIC PHI AND
C     DPHIS2 ARE ZERO.
C
      PHI(1) = (0.,0.)
      DPHIS2(1) = (0.,0.)
C
C
      AA = RM*PI/A4
      DPHIS1(1) = CMPLX(AA,0.)
      DPHIDR = A4*DPHIS1(1)
      DPHIDX = A2*DPHIS1(1)
      CPR(1) = -2.*REAL(DPHIDX)
      CPI(1) = -2.*AIMAG(DPHIDX)
C
      IF ( IPRINT .NE. 1 ) RETURN
      WRITE(6,90)
      WRITE(6,95)
      WRITE(6,100) X(1),DPHIS1(1),DPHIS2(1),PHI(1),
     1   R(1),DPHIDR,DPHIDX
      WRITE(6,110) CPR(1),CPI(1)
      RETURN
   90 FORMAT('1',T10,'X',T22,'D(PHI)/D(S1)',T46,
     1   'D(PHI)/D(S2)',T70,'PHI')
   95 FORMAT(' ',T10,'R',T23,'D(PHI)/DR',T47,'D(PHI)/DX',/)
  100 FORMAT(3H I ,2X,F8.4,4X,3(F12.5,F9.5,3X),/' ',4X,F8.4,
     1   4X,3(F12.5,F9.5,3X))
  110 FORMAT('0',10X,'INITIAL VALUE:  INNER SURFACE OF',
     1   ' OUTER CYLINDER',/' ',30X,'CP(REAL) = ',F8.4,10X,
     2   'CP(IMAG) = ',F8.4,//)
      END


C                  THIS IS SUBROUTINE COMPXR
C
      SUBROUTINE COMPXR
C
C     SUBROUTINE COMPXR COMPUTES THE (X,R) COORDINATES OF A
C     GRID POINT.
C
      DIMENSION X(400),R(400),XPT(400),CPR(400),CPI(400)
      COMMON/BLCK1/ FSTRMN,DELTAS,DX,DH,DSTSTR,RO,RI,REDFRQ
      COMMON/BLCK3/ NGRDFN,NPTS,LCOUNT,IHAVEP,N,M,M1,IPRINT
      COMMON/BLCK5/ X,R,XPT,CPR,CPI
C
      I = IHAVEP + 1
      J = IHAVEP
C
C     IF THIS IS THE FIRST POINT ON A NEW RIGHT RUNNING MACH
C     LINE, GO TO 1.
C
      IF ( J .EQ. 0 ) GO TO 1
C
      X(I) = X(J) + DX
      R(I) = R(J) -DH
      RETURN
C
    1 R(I) = RO
      X(I) = X(1) + DSTSTR
      RETURN
```

```
      END


C                  THIS IS SUBROUTINE MACHLN
C
      SUBROUTINE MACHLN
C
C     SUBROUTINE MACHLN COMPUTES THE VALUES OF THE FLOW
C     FIELD QUANTITIES ALONG THE INITIAL RIGHT-RUNNING MACH
C     LINE WHERE PHI AND DPHIS2 ARE ZERO.
C
      COMPLEX A,C
      COMPLEX DPHIDR,DPHIDX
      COMPLEX PHI(400),DPHIS1(400),DPHIS2(400)
      DIMENSION X(400),R(400),XPT(400),CPR(400),CPI(400)
      COMMON/BLCK1/ FSTRMN,DELTAS,DX,DH,DSTSTR,RO,RI,REDFRQ
      COMMON/BLCK2/ A1,A2,A3,A4,RN,RM,RM1,FACT1,FACT2
      COMMON/BLCK3/ NGRDFN,NPTS,LCOUNT,IHAVEP,N,M,M1,IPRINT
      COMMON/BLCK4/ PHI,DPHIS1,DPHIS2
      COMMON/BLCK5/ X,R,XPT,CPR,CPI
C
      I = IHAVEP + 1
      J = IHAVEP
C
C     CALCULATE THE COEFFICIENTS ASSOCIATED WITH THE RIGHT-
C     RUNNING MACH LINE.
C     LINE.
C
      AA1 = 1. - A1/R(I)
      A = CMPLX(AA1,A3)
      C1 = 1. + A1/R(J)
      C = DPHIS1(J)*CMPLX(C1,-A3)
C
      PHI(I) = (0.,0.)
      DPHIS2(I) = (0.,0.)
      DPHIS1(I) = C/A
C
      DPHIDR =  A4*DPHIS1(I)
      DPHIDX = A2*DPHIS1(I)
C
      IF ( IPRINT .NE. 1 ) RETURN
      WRITE(6,100) X(I),DPHIS1(I),DPHIS2(I),PHI(I),
     1   R(I),DPHIDR,DPHIDX
      WRITE(6,111)
      RETURN
  100 FORMAT(3H M ,2X,F8.4,4X,3(F12.5,F9.5,3X),/' ',4X,F8.4,
     1    4X,3(F12.5,F9.5,3X))
  111 FORMAT('0')
      END


C                   THIS IS SUBROUTINE RAD1
C
C     THIS IS SUBROUTINE RAD1
      SUBROUTINE RAD1
C
C     SUBROUTINE RAD1 COMPUTES THE FLOW FIELD QUANTITIES AT
C     THE SURFACE OF THE INNER RADIUS WHERE THE BOUNDARY
C     CONDITION IS  DEPENDENT ON N , THE CIRCUMFERENTIAL
C     MODE NUMBER.  IF  N  IS:
C        EVEN  --  DPHIDR = 0, AND THUS DPHIS1 = DPHIS2.
C        ODD   --  CP = 0.
C
      COMPLEX A,B,C,TANR,DPHIDR,DPHIDX,DELTA,GAMMA,IK
      COMPLEX PHI(400),DPHIS1(400),DPHIS2(400)
      DIMENSION X(400),R(400),XPT(400),CPR(400),CPI(400)
      COMMON/BLCK1/ FSTRMN,DELTAS,DX,DH,DSTSTR,RO,RI,REDFRQ
      COMMON/BLCK2/ A1,A2,A3,A4,RN,RM,RM1,FACT1,FACT2
```

```
      COMMON/BLCK3/ NGRDFN,NPTS,LCOUNT,IHAVEP,N,M,M1,IPRINT
      COMMON/BLCK4/ PHI,DPHIS1,DPHIS2
      COMMON/BLCK5/ X,R,XPT,CPR,CPI
C
      I = IHAVEP + 1
      J = IHAVEP
C
C     CALCULATE THE COEFFICIENTS ASSOCIATED WITH THE RIGHT-
C     RUNNING MACH LINE.
C
      AA1 = 1. - A1/R(I)
      A = CMPLX(AA1,A3)
      B1 = A1/R(I) - 0.25*DELTAS*DELTAS*(FACT1 -
     1   FACT2/(R(I)*R(I)))
      B = CMPLX(B1,A3)
      C1 = 1. + A1/R(J)
      C2 = -A1/R(J) + 0.25*DELTAS*DELTAS*(FACT1 -
     1   FACT2/(R(I)*R(I)))
      C3 = (2.*FACT1 - FACT2/(R(J)*R(J)) -
     1   FACT2/(R(I)*R(I)))*DELTAS*0.5
      C = DPHIS1(J)*CMPLX(C1,-A3) + DPHIS2(J)*CMPLX(C2,-A3)
     1    + PHI(J)*CMPLX(C3,0.)
C
      IF ( MOD(N,2) .EQ. 0 ) GO TO 1
C
C     APPLY THE BOUNDARY CONDITION FOR ODD VALUES OF N.
C     CP = (IK)PHI + DPHIDX = 0.  USE THIS B.C., THE FINITE
C     DIFFERENCE EQN. FOR DPHIDX, AND THE EQN. FOR THE RIGHT
C     RUNNING MACHLINE TO SOLVE FOR DPHIS1 AND DPHIS2.
C
      DELTA = A2 + 0.5*IK*DELTAS
      GAMMA = IK*( PHI(J) + 0.5*DELTAS*DPHIS2(J) )
C
C     SOLVING FOR DPHIS1 AND DPHIS2
C
      DPHIS1(I) = ( C + B*GAMMA/DELTA )/( A - B*A2/DELTA )
      DPHIS2(I) = ( C - A*DPHIS1(I) )/B
C
      DPHIDX = A2*( DPHIS1(I) + DPHIS2(I) )
      DPHIDR = A4*( DPHIS1(I) - DPHIS2(I) )
      GO TO 2
C
C     APPLY THE BOUNDARY CONDITION FOR EVEN VALUES OF  N.
C
    1 DPHIS1(I) = C/(A+B)
      DPHIS2(I) = DPHIS1(I)
      DPHIDR = (0.,0.)
      DPHIDX = A2*(DPHIS1(I) + DPHIS2(I))
C
C     PHI.
C
    2 PHI(I) = PHI(J) + 0.5*(DPHIS2(J) + DPHIS2(I))*DELTAS
C
      IF ( IPRINT .NE. 1 ) RETURN
      WRITE(6,100) X(I),DPHIS1(I),DPHIS2(I),PHI(I),
     1   R(I),DPHIDR,DPHIDX
      RETURN
  100 FORMAT(3H RI,2X,F8.4,4X,3(F12.5,F9.5,3X),/' ',4X,F8.4,
     1   4X,3(F12.5,F9.5,3X))
      END


C                     THIS IS SUBROUTINE RAD2
C
      SUBROUTINE RAD2
C
C     SUBROUTINE RAD2 COMPUTES THE FLOW FIELD QUANTITIES AT
C     THE INNER SURFACE OF THE OUTER CYLINDER, WHERE THE
C     FLOW TANGENCY CONDITION PRESCRIBES DPHIDR.
C
```

```
      COMPLEX D,E,F,TANR
      COMPLEX DPHIDR,DPHIDX
      COMPLEX PHI(400),DPHIS1(400),DPHIS2(400)
      DIMENSION X(400),R(400),XPT(400),CPR(400),CPI(400)
      COMMON/BLCK1/ FSTRMN,DELTAS,DX,DH,DSTSTR,RO,RI,REDFRQ
      COMMON/BLCK2/ A1,A2,A3,A4,RN,RM,RM1,FACT1,FACT2
      COMMON/BLCK3/ NGRDFN,NPTS,LCOUNT,IHAVEP,N,M,M1,IPRINT
      COMMON/BLCK4/ PHI,DPHIS1,DPHIS2
      COMMON/BLCK5/ X,R,XPT,CPR,CPI
      DATA PI/3.141593/
C
      I = IHAVEP + 1
      J = IHAVEP + 2
      K = LCOUNT
      ARG = RM*PI*X(I)
C
C
C     CALCULATE THE COEFFICIENTS ASSOCIATED WITH THE LEFT
C     RUNNING MACH LINE.
C
      D1 = A1/R(I) + 0.25*DELTAS*DELTAS*(FACT1 - FACT2/(R(I)
     1  *R(I)))
      D = CMPLX(D1,-A3)
      E1 = -1. - A1/R(I)
      E = CMPLX(E1,-A3)
      F1 = -A1/R(J) - 0.25*DELTAS*DELTAS*(FACT1 - FACT2/
     1   (R(I)*R(I)))
      F2 = -1. + A1/R(J)
      F3 = (2.*FACT1 - FACT2/(R(I)*R(I)) - FACT2/(R(J)*
     1   R(J)))*DELTAS*0.5
      F = DPHIS1(J)*CMPLX(F1,A3) + DPHIS2(J)*CMPLX(F2,A3) -
     1   PHI(J)*CMPLX(F3,0.)
C
C     CALCULATE THE FLOW TANGENCY CONDITION  -  TANR
C
C     BOUNDARY CONDITIONS FOR PANEL FLUTTER
C
      TANR1 = RM*PI*COS(ARG)
      TANR2 = REDFRQ*SIN(ARG)
      TANR = CMPLX(TANR1,TANR2)
C
      DPHIS1(I) = ( F + TANR*E/A4 )/( D + E )
      DPHIS2(I) = ( F - TANR*D/A4 )/( D + E )
      DPHIDR = A4*(DPHIS1(I) - DPHIS2(I))
      DPHIDX = A2*(DPHIS1(I) + DPHIS2(I))
C
C     INTEGRATE ALONG THE LEFT RUNNING MACH LINE TO FIND PHI
C
      PHI(I) = PHI(J) + 0.5*(DPHIS1(J) + DPHIS1(I))*DELTAS
C
C     CALCULATE THE PRESSURE COEFFICIENT AT THIS GRID POINT.
C
      XPT(K) = X(I)
      CPR(K) = -2.*(-REDFRQ*AIMAG(PHI(I)) + REAL(DPHIDX))
      CPI(K) = -2.*(REDFRQ*REAL(PHI(I)) + AIMAG(DPHIDX))
C
      IF ( IPRINT .NE. 1 ) RETURN
      WRITE(6,100) X(I),DPHIS1(I),DPHIS2(I),PHI(I),
     1   R(I),DPHIDR,DPHIDX,TANR
      WRITE(6,110) CPR(K),CPI(K)
      RETURN
  100 FORMAT(3H RO,2X,F8.4,4X,3(F12.5,F9.5,3X),/' ',4X,F8.4,
     1   4X,3(F12.5,F9.5,3X))
  110 FORMAT('0',10X,'INNER SURFACE OF OUTER CYLINDER',/' ',
     1   30X,'CP(REAL) = ',F8.4,10X,'CP(IMAG) = ',F8.4,//)
      END
C               THIS IS SUBROUTINE GENFPT
C
      SUBROUTINE GENFPT(IER)
C
C     SUBROUTINE GENFPT COMPUTES THE FLOW FIELD QUANTITIES
C     AT A GENERAL FLOW FIELD POINT.
```

```
C
      COMPLEX PHIL
      COMPLEX A,B,C,D,E,F,DENOM
      COMPLEX DPHIDR,DPHIDX
      COMPLEX PHI(400),DPHIS1(400),DPHIS2(400)
      DIMENSION X(400),R(400),XPT(400),CPR(400),CPI(400)
      COMMON/BLCK1/ FSTRMN,DELTAS,DX,DH,DSTSTR,RO,RI,REDFRQ
      COMMON/BLCK2/ A1,A2,A3,A4,RN,RM,RM1,FACT1,FACT2
      COMMON/BLCK3/ NGRDFN,NPTS,LCOUNT,IHAVEP,N,M,M1,IPRINT
      COMMON/BLCK4/ PHI,DPHIS1,DPHIS2
      COMMON/BLCK5/ X,R,XPT,CPR,CPI
C
      I = IHAVEP + 1
      J = IHAVEP
      K = IHAVEP + 2
C
C     COMPUTE THE COEFFICIENTS OF DPHIS1 AND DPHIS2.
      FACT3 = FACT2/(R(I)*R(I))
      FACT4 = FACT2/(R(K)*R(K))
      FACT5 = FACT2/(R(J)*R(J))
C
C     A,B AND C ARE ASSOCIATED WITH THE RIGHT RUNNING MACH
C     LINE, WHILE D,E AND F ARE ASSOCIATED WITH THE LEFT.
C
      AA1 = 1. - A1/R(I)
      A = CMPLX(AA1,A3)
      B1 = -AA1 + 1. - 0.25*DELTAS*DELTAS*(FACT1 - FACT3)
      B = CMPLX(B1,A3)
      C1 = 1. + A1/R(J)
      C2 = -C1 + 1. + 0.25*DELTAS*DELTAS*(FACT1 - FACT3)
      C3 = (2.*FACT1 - FACT3 - FACT5)*DELTAS*0.5
      C = DPHIS1(J)*CMPLX(C1,-A3) + DPHIS2(J)*CMPLX(C2,-A3)
     1   + PHI(J)*CMPLX(C3,0.)
C
      D1 = A1/R(I) + 0.25*DELTAS*DELTAS*(FACT1 - FACT3)
      D = CMPLX(D1,-A3)
      E1 = -1. - A1/R(I)
      E = CMPLX(E1,-A3)
      F1 = -A1/R(K) - 0.25*DELTAS*DELTAS*(FACT1 - FACT3)
      F2 = -1. + A1/R(K)
      F3 = (2.*FACT1 - FACT3 - FACT4)*DELTAS*0.5
      F = DPHIS1(K)*CMPLX(F1,A3) + DPHIS2(K)*CMPLX(F2,A3) -
     1   PHI(K)*CMPLX(F3,0.)
C
C     USE CRAMERS RULE TO SOLVE FOR DPHIS1 AND DPHIS2
C     PROVIDED THAT THE DETERMINANT OF COEFFICIENTS IS NON-
C     SINGULAR.
C
      DENOM = A*E - D*B
      IF ( CABS( DENOM ) .LT. 1.E-05 ) GO TO 1
      DPHIS1(I) = (C*E - B*F)/DENOM
      DPHIS2(I) = (A*F - C*D )/DENOM
      DPHIDR = A4*(DPHIS1(I) - DPHIS2(I))
      DPHIDX = A2*(DPHIS1(I) + DPHIS2(I))
C
C     INTEGRATE ALONG THE RIGHT RUNNING MACH LINE TO FIND
C     PHI.
C
      PHI(I) = PHI(J) + 0.5*(DPHIS2(I) + DPHIS2(J))*DELTAS
      PHIL = PHI(K) + 0.5*(DPHIS1(K) + DPHIS1(I))*DELTAS
C
      IF ( IPRINT .NE. 1 ) RETURN
      WRITE(6,100) X(I),DPHIS1(I),DPHIS2(I),PHI(I),
     1   R(I),DPHIDR,DPHIDX,PHIL
      WRITE(6,111)
      RETURN
C
    1 IER = 1
      WRITE(6,110) DENOM
      RETURN
  100 FORMAT(3H G ,2X,F8.4,4X,3(F12.5,F9.5,3X),/' ',4X,F8.4,
     1   4X,3(F12.5,F9.5,3X))
```

```
  110 FORMAT('0',///,' ',35X,'ABNORMAL TERMINATION: ',
     1   'DETERMINANT OF COEFFICIENTS IS SINGULAR',/'0',50X,
     2   'DENOMINATOR = ',E12.5)
  111 FORMAT('0')
      END



C                     THIS IS SUBROUTINE QSF
C          SUBROUTINE QSF
      SUBROUTINE QSF(H,Y,Z,NDIM)
C             CALL QSF (H,Y,Z,NDIM)
C          DESCRIPTION OF PARAMETERS
C             H        - THE INCREMENT OF ARGUMENT VALUES.
C             Y        - THE INPUT VECTOR OF FUNCTION VALUES.
C             Z        - THE RESULTING VECTOR OF INTEGRAL
C                        VALUES.  Z  MAY BE IDENTICAL WITH  Y.
C                        IDENTICAL WITH Y.
C             NDIM     - THE DIMENSION OF VECTORS Y AND Z.
C
C          REMARKS
C             NO ACTION IN CASE NDIM LESS THAN 3.
C
C
C
      DIMENSION Y(1),Z(1)
C
      HT=.3333333*H
      IF(NDIM-5)7,8,1
C
C     NDIM IS GREATER THAN 5. PREPARATIONS OF INTEGRATION
C     LOOP
C
    1 SUM1=Y(2)+Y(2)
      SUM1=SUM1+SUM1
      SUM1=HT*(Y(1)+SUM1+Y(3))
      AUX1=Y(4)+Y(4)
      AUX1=AUX1+AUX1
      AUX1=SUM1+HT*(Y(3)+AUX1+Y(5))
      AUX2=HT*(Y(1)+3.875*(Y(2)+Y(5))+2.625*(Y(3)+Y(4))+
     1   Y(6))
      SUM2=Y(5)+Y(5)
      SUM2=SUM2+SUM2
      SUM2=AUX2-HT*(Y(4)+SUM2+Y(6))
      Z(1)=0.
      AUX=Y(3)+Y(3)
      AUX=AUX+AUX
      Z(2)=SUM2-HT*(Y(2)+AUX+Y(4))
      Z(3)=SUM1
      Z(4)=SUM2
      IF(NDIM-6)5,5,2
C
C     INTEGRATION LOOP
    2 DO 4 I=7,NDIM,2
      SUM1=AUX1
      SUM2=AUX2
      AUX1=Y(I-1)+Y(I-1)
      AUX1=AUX1+AUX1
      AUX1=SUM1+HT*(Y(I-2)+AUX1+Y(I))
      Z(I-2)=SUM1
      IF(I-NDIM)3,6,6
    3 AUX2=Y(I)+Y(I)
      AUX2=AUX2+AUX2
      AUX2=SUM2+HT*(Y(I-1)+AUX2+Y(I+1))
    4 Z(I-1)=SUM2
    5 Z(NDIM-1)=AUX1
      Z(NDIM)=AUX2
      RETURN
    6 Z(NDIM-1)=SUM2
      Z(NDIM)=AUX1
      RETURN
```

```
C        END OF INTEGRATION LOOP
C
       7 IF(NDIM-3)12,11,8
C
C        NDIM IS EQUAL TO 4 OR 5
       8 SUM2=1.125*HT*(Y(1)+Y(2)+Y(2)+Y(2)+Y(3)+Y(3)+Y(3)+
        1   Y(4))
         SUM1=Y(2)+Y(2)
         SUM1=SUM1+SUM1
         SUM1=HT*(Y(1)+SUM1+Y(3))
         Z(1)=0.
         AUX1=Y(3)+Y(3)
         AUX1=AUX1+AUX1
         Z(2)=SUM2-HT*(Y(2)+AUX1+Y(4))
         IF(NDIM-5)10,9,9
       9 AUX1=Y(4)+Y(4)
         AUX1=AUX1+AUX1
         Z(5)=SUM1+HT*(Y(3)+AUX1+Y(5))
      10 Z(3)=SUM1
         Z(4)=SUM2
         RETURN
C
C        NDIM IS EQUAL TO 3
      11 SUM1=HT*(1.25*Y(1)+Y(2)+Y(2)-.25*Y(3))
         SUM2=Y(2)+Y(2)
         SUM2=SUM2+SUM2
         Z(3)=HT*(Y(1)+SUM2+Y(3))
         Z(1)=0.
         Z(2)=SUM1
      12 RETURN
         END
```

LIST OF REFERENCES

1. Bisplinghoff, R. L., Ashley, H., and Halfman, R. L., Aeroelasticity, p 527-545, Addison-Wesley Publishing Company, Inc., 1955.

2. Brix, C. W., A Study of Supersonic Flow Past Vibrating Shells and Cascades, Aeronautical Engineer's Thesis, Naval Post Graduate School, June, 1973.

3. Brix, C. W., and Platzer, M. F., Theoretical Investigation of Supersonic Flow Past Oscillating Cascades with Subsonic Leading-Edge Locus, paper presented at AIAA 12th Aerospace Sciences Meeting, Washington D. C., January 30-February 1, 1974.

4. Carta, F. O., "Coupled Blade-Disk-Shroud Flutter Instabilities in Turbojet Engine Roters", Transactions of the ASME, Journal of Engineering for Power, v. 89(A), No. 3, p 419-426, July 1967.

5. Chalkley, H. G., A Study of Supersonic Cascade Flutter, Aeronautical Engineer's Thesis, Naval Post Graduate School, June, 1972.

6. Detroit Diesel Allison Report, Aeroelasticity in Turbomachines: Proceedings of a Workshop Held on June 1-2, 1972, S. Fleeter (editor).

7. Fleeter, S., McClure, R. B., Sinnet, G. T., and Holtman, R. L., A Unique Supersonic Inlet Unsteady Aerodynamic Cascade Experiment, paper presented at AIAA 8th Aerodynamic Testing Conference, Bethesda, Maryland, July 8-10, 1974.

8. Fung, Y. C., An Introduction to the Theory of Aeroelasticity, p. 210-244, Dover Publications, Inc., 1969

9. Garrick, I. E., and Rubinow, S. J., "Flutter and Oscillating Air-Force Calculations for an Airfoil in Two-Dimensional Supersonic Flow", NACA Report 846, 1946.

10. Gorelov, D. N., "Lattice Plates in an Unsteady Supersonic Flow", Mekhanika Zhidkosti i Gaza, v. 1, No. 4, p. 50-58, 1966.

11. Kurosaka, M., "On the Unsteady Supersonic Cascade with a Subsonic Leading Edge - An Exact First-Order Theory, Part 1 and 2", Transactions of the ASME, Journal of Engineering for Power, v. 96, No. 1, p. 13-31, January, 1974.

12. Kurosaka, M., Supersonic Cascade Study, work currently performed for the Air Force Office of Scientific Research.

13. Kurasaka, M., "On the Issue of Resonance in the Subsonic Leading Edge Locus Problem in an Unsteady Supersonic Cascade", *General Electric Technical Information Series*, Document No. 74 CRD293, November, 1974.

14. Lane, F., "System Mode Shapes in the Flutter of Compressor Blade Rows", *Journal of Aeronautical Sciences*, v. 23, p. 54-66, January, 1956.

15. Nagashima, T., and Whitehead, D. J., *Aerodynamic Forces and Moments for Vibrating Supersonic Cascade Blades*, CUED/A-Turb./TR-59, Cambridge University, 1974.

16. Owczarek, J. A., *Fundamentals of Gas Dynamics*, p. 278-357, International Text Book Company, 1964.

17. Platzer, M. F., Brix, C. W., and Webster, K. A., "A Linearized Characteristics Method for Supersonic Flow Past Vibrating Shells", *AIAA Journal*, v. 11, No.9, September, 1973.

18. Pratt and Whitney Aircraft Report No. PWA 5142, *Supersonic Chordwise Bending Flutter in Cascades*, by L. E. Snyder, November, 1974.

19. Snyder, L. E., and Commerford, G. L., *Supersonic Unstalled Flutter in Fan Rotors; Analytical and Experimental Results*, ASME paper 74-GT-40 presented at the ASME Gas Turbine Conference, Zurich, Switzerland, March 31-April 4, 1974.

20. Stevens Institute of Technology, *Research on the Flutter of Axial Turbomachinery Blading*, by F. Sisto and R. H. Ni, Technical Report ME-RT 74-008, May, 1974.

21. Teipel, I., "Ein Charakteristikenverfahren zur Berchnung der Verallgemeinerten ebenen Flatterluftkrafte", *Zeitschrift fur Flugwissenschaften*, v. 10, p. 374-379, October, 1962.

22. Theodorsen, T., "General Theory of Aerodynamic Instability and the Mechanism of Flutter", *NACA Report 496*, 1935.

23. Verdon, J. M., "The Unsteady Aerodynamics of a Finite Supersonic Cascade with Subsonic Axial Flow", *Transactions of the ASME,Journal of Applied Mechanics*, v. 40, No. 3, p. 667-671, September, 1973.

24. Verdon, J. M., and McCune, J. E., *The Unsteady Supersonic Cascade in Supersonic Axial Flow*, AIAA paper 75-22 presented at the AIAA 13th Aerospace Sciences Meeting, Pasadena, California, January 20-22, 1975.

25. United Aircraft Research Laboratories Report N210662-2, *Unsteady Aerodynamic Analysis of a Supersonic Cascade in Subsonic Axial Flow*, by J. M. Verdon, April, 1974.

26. Verdon, J. M., private communication, May, 1975.

27. Widnall, S. E., and Dowell, E. H., "Aerodynamic Forces on an Oscillating Cylindrical Duct with an Internal Flow", *Journal of Sound and Vibration*, v. 6, No. 1, p. 71-85, 1967.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor R. W. Bell, Code 57Be Department of Aeronautics Naval Postgraduate School Monterey, California 93940	2
4.	Assoc. Professor M. F. Platzer, Code 57Pl Department of Aeronautics Naval Postgraduate School Monterey, California 93940	15
5.	Professor T. H. Gawain, Code 57Gn Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
6.	Vist. Assoc. Professor R. P. Shreeve, Code 57Sf Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
7.	Dr. H. J. Mueller (AIR - 310) Naval Air Systems Command Washington, D. C. 20360	1

8. LT J. K. Bell 2
125 Corpus Christi Road
Alameda, California 94501

9. Dr. J. Verdon 1
United Aircraft Research Laboratories
East Hartford, Connecticut 06108

10. Dr. S. Fleeter 1
Detroit Diesel Allison
Division of General Motors
Indianapolis, Indiana 46206

11. Dr. W. R. Chadwick 1
Naval Surface Weapons Laboratory
Dahlgren, Virginia 22448

161363
Thesis
B36173 Bell
c.1 Theoretical investigation of the flutter characteristics of supersonic cascades with subsonic leading-edge locus.